VEE Pro Benutzerhandbuch

Hinweis

Informationen zu Warenzeichen

Microsoft®, MS-DOS®, Windows®, MS Windows®, and Windows NT® sind in den USA und anderen Ländern eingetragene Warenzeichen der Microsoft Corporation.

MATLAB® ist ein eingetragenes Warenzeichen von The MathWorks, Inc.

Netscape ist ein in den USA und anderen Ländern eingetragenes Warenzeichen der Netscape Communications Corporation.

UNIX® ist ein eingetragenes Warenzeichen der Open Group.

Druckhistorie

1. Ausgabe - März 2000 Entspricht Software-Version 6.0

In diesem Handbuch verwendete Konventionen

In diesem Handbuch werden folgende typografische Konventionen verwendet:

Erste Schritte	Text in Kursivschrift wird für Handbuchtitel und zur Hervorhebung verwendet.
Dialogfenster	Text in Fettschrift wird für das erste Vorkommen eines Begriffs verwendet, der im Glossar erläutert wird.
`Datei`	Text in Computerschrift kennzeichnet Text, der auf dem Bildschirm angezeigt wird, z. B. Menünamen, Funktionen, Schaltflächen oder Text, den Sie eingeben müssen.
`dir` *dateiname*	In diesem Kontext steht die Computerschrift für ein Argument, das Sie genau wie angegeben eintippen müssen. Der Text in Kursivschrift steht für ein Argument, das Sie durch einen tatsächlichen Wert ersetzen.
`File ⇒ Open`	Das Zeichen "⇒" wird als Kurzschrift zur Darstellung der Position der Agilent VEE-Funktionen im Menü verwendet. "`File ⇒ Open`" bedeutet beispielsweise, dass Sie das Menü `File` und anschließend die Funktion `Open` auswählen müssen.
`Sml | Med | Lrg`	Auswahloptionen in Computerschrift, die durch einen senkrechten Strich (|) getrennt sind, geben an, dass Sie eine dieser Optionen auswählen müssen.

Drücken Sie die **Eingabetaste**	In diesem Kontext kennzeichnet die Fettschrift eine Taste auf der Tastatur.
Drücken Sie **Strg + O**	Steht für eine Kombination von Tasten auf der Tastatur, die gleichzeitig gedrückt werden müssen.

Inhaltsverzeichnis

2. Agilent VEE Programmiertechniken

3. Einfache Methoden zum Steuern von Instrumenten

7. Integrieren von Programmen in anderen Sprachen

8. Verwenden von Agilent VEE-Funktionen

9. Sequenzieren von Tests

10. Verwenden von Bedienerschnittstellen

11. Optimieren von Agilent VEE-Programmen

12. Plattformspezifische Faktoren und Web-Überwachung

A. Zusätzliche Übungen

Glossar

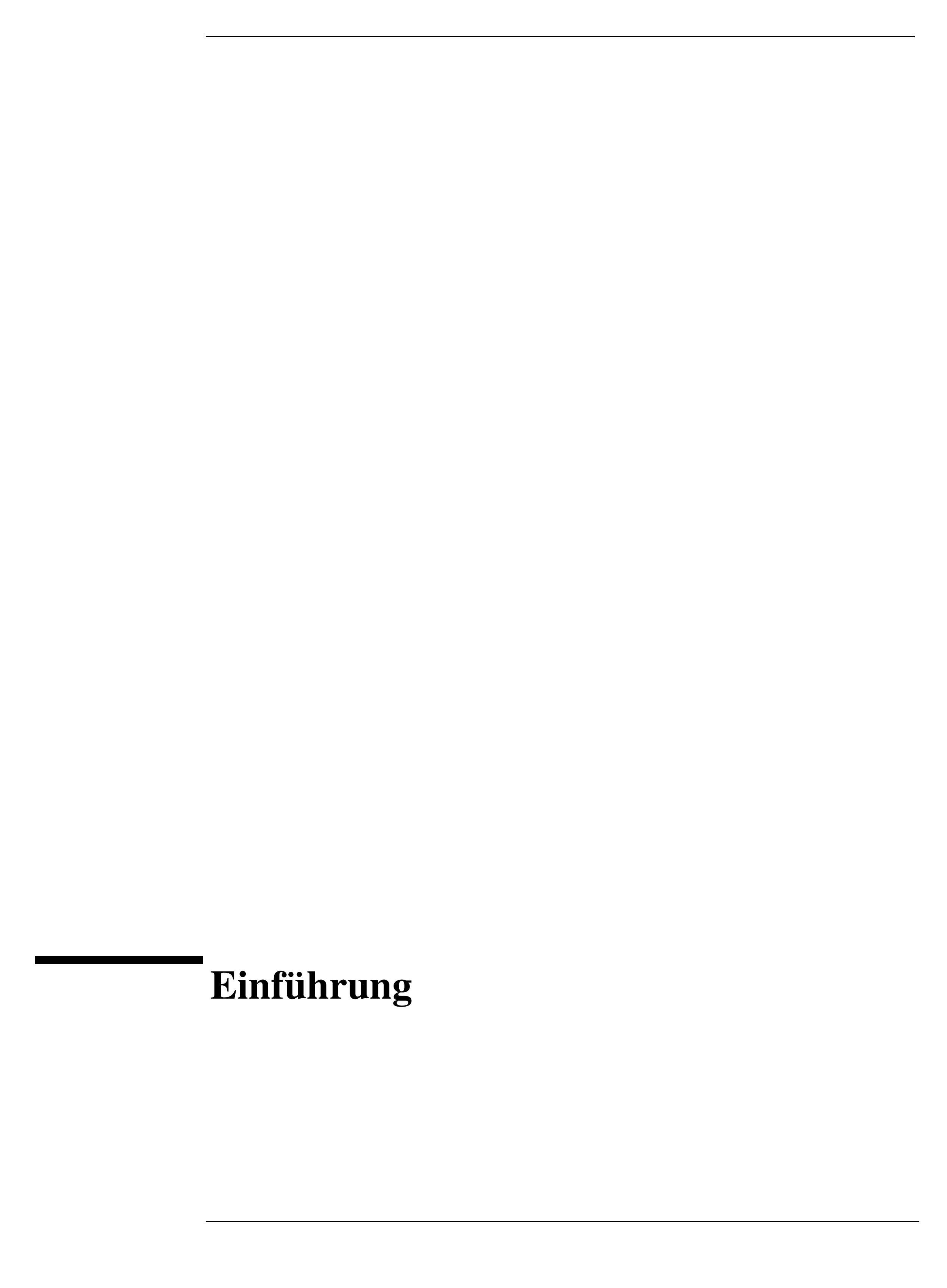
Index



Abbildungsverzeichnis

Tabellenverzeichnis





Einführung

Einführung

Dieses Kapitel stellt Agilent VEE und seine wichtigsten Funktionen vor. Außerdem erfahren Sie, wie Sie VEE installieren, Informationen dazu erhalten und Support zu VEE anfordern können.

Überblick über Agilent VEE

Agilent VEE ist eine grafische Programmiersprache, die zum Erstellen von Anwendungen im Bereich Test und Messtechnik sowie von Programmen mit Bedienerschnittstellen optimiert wurde. Diese Version der Agilent VEE Produktfamilie umfasst VEE Pro 6.0 für Gruppen von Technikern, die komplexe Test- und Messsysteme erstellen, und VEE OneLab 6.0 für einzelne Techniker und Wissenschaftler, die für die Konzeption und Beschaffung von Daten zuständig sind.

Vorteile von Agilent VEE bei der Testentwicklung

VEE bietet eine Vielzahl von Vorteilen bei der Testentwicklung:

- Erhebliche Steigerung der Produktivität. Von Anwendern erfahren wir, dass die Zeit für die Programmentwicklung um bis zu 80% zurückgegangen ist.

- Einsatz von VEE in einem breiten Spektrum von Anwendungen einschließlich Funktionstest, Design-Überprüfung, Kalibrierung, Datenbeschaffung und Steuerung.

- Erzielen einer hohen E/A-Flexibilität bei Instrumenten zur Steuerung von GPIB, VXI, Seriellen Verbindungen, GPIO, PC-Zusatzkarten sowie LAN-Instrumenten. Verwenden von “Panel”-Treibern, VXI*Plug&Play*-Treibern, ODAS-Treibern, “Direkt-E/A” über Standardschnittstellen oder importierten Bibliotheken anderer Hersteller.

- Verwenden von ActiveX-Automatisierungs- und Steuerelementen an PCs zur Steuerung anderer Anwendungen wie MS Word, Excel und Access zur Unterstützung beim Generieren von Berichten, beim Anzeigen und der Analyse von Daten sowie beim Ablegen von Testergebnissen in einer Datenbank zur späteren Verwendung.

- Verbesserung des Durchsatzes, einfacheres Erstellen größerer Programme und größere Flexibilität bei der Verwaltung von Instrumenten. VEE enthält einen Compiler, eine professionelle Entwicklungsumgebung für große, komplexe Programme sowie erweiterte Funktionen zur Instrumentenverwaltung.

- Nutzen der Investitionen in Textsprachen wie C/C++, Visual Basic, Pascal, Fortran und Rocky Mountain Basic.

Erstellen von Programmen in Agilent VEE

VEE-Programme werden durch Auswählen von Objekten in Menüs und Verbinden dieser Objekte erstellt. Das Ergebnis in VEE erinnert an ein Datenflussdiagramm; diese Darstellung ist einfacher zu verstehen und zu verwenden als traditionelle Codezeilen. Mit VEE entfällt der aufwändige Zyklus Bearbeiten-Kompilieren-Linken-Ausführen.

Die beiden folgenden Abbildungen vergleichen eine einfache Funktion, die zunächst in einer Textsprache (ANSI C) und anschließend in VEE programmiert wurde. In beiden Fällen erstellt die Funktion einen "Array" von 10 Zufallszahlen, sucht den größten Wert und zeigt den Array und den größten Wert an.

Abbildung I-1 zeigt das Programm “Random” in der Textsprache ANSI C.

```
/* Programm zum Ermitteln des größten Elements in
einem Array */

#include <math.h>

main( )

{

double num[10],max;

int i;

for (i=0;i<10,i++){

num[i]=(double) rand( )/pow(2.0,15.0);

printf("%f/n",num[i];

}

max=num[0];

for {i=1;i<10;i++){

if (num[i]>max)max=num[i];

}

printf("/nmax; %f/n",max);

}
```

Abbildung I-1. Das Programm “Random” in ANSI C

Abbildung I-2 zeigt das gleiche Programm in VEE.

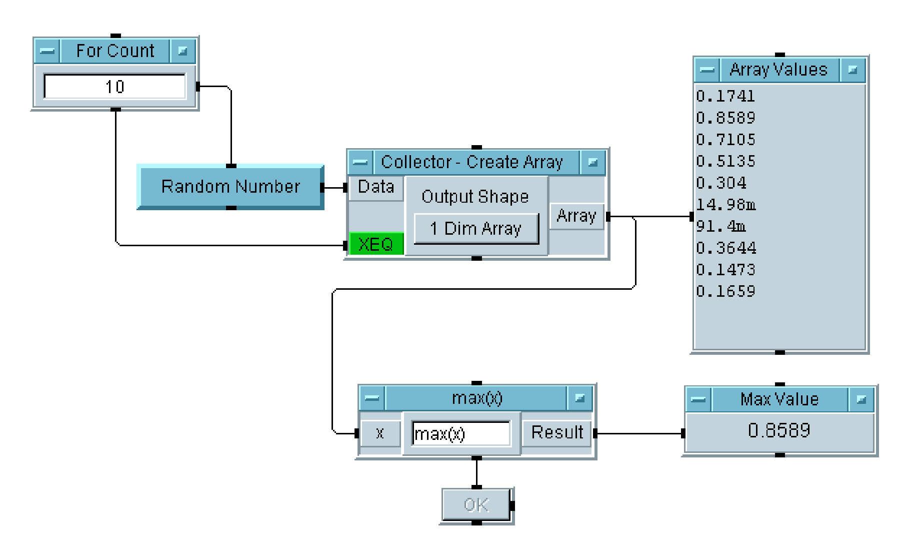


Abbildung I-2. Das gleiche Programm “Random” in VEE

In VEE wird das Programm mit Programmelementen, den so genannten **Objekten**, erstellt. Objekte sind die Bausteine eines VEE-Programms. Sie führen verschiedene Funktionen aus wie beispielsweise E/A-Operationen, Analyse und Anzeige. Wenn Sie die Objekte mit allen ihren Verbindungen anzeigen, wie in Abbildung I-2 dargestellt, wird dies als **Detailansicht** ("detail view") bezeichnet. Die Detailansicht ist analog zum Quellencode in einer Textsprache.

In VEE wandern Daten in konsistenter Weise von einem Objekt zum nächsten: Dateneingang links, Datenausgang rechts, operative Pin-Verbindungen oben und unten.

Durch Verbinden der Objekte wird ein Programm gebildet. Verfolgen Sie das Programm von links nach rechts. Im Programm “Random” (siehe Abbildung I-2) wird dem Objekt `Collector - Create Array` zehn Mal eine Zufallszahl hinzugefügt und somit ein Array erstellt. Das Programm sucht anschließend den größten Wert in dem Array und zeigt den größten Wert (`Max Value`) und die Array-Werte (`Array Values`) an.

Mit VEE und seinem modularen Ansatz reduzieren Sie die erforderliche Zeit zum Erstellen von Programmen für die Steuerung von Instrumenten, generieren angepasste Datenanzeigen und entwickeln Bedienerschnittstellen. Diese Methode der Testentwicklung ermöglicht eine deutlich größere Produktivitätssteigerung als herkömmliche Techniken.

Hinweis

In Abbildung I-2 werden manche Objekte detailliert angezeigt, andere dagegen nur als Name. Die ausführlich dargestellten Objekte werden in einer **offenen Ansicht** ("open view") angezeigt. Die offene Ansicht ermöglicht die Anzeige von Details zu dem Objekt. Sie können Objekte **als Symbol anzeigen**, um Speicherplatz zu sparen und das Programm schneller auszuführen, oder Sie können Objekte reduzieren, sodass nur ihre Namen angezeigt werden.

In Abbildung I-2 wird beispielsweise das Objekt mit der Beschriftung `Random Number` als Symbol angezeigt. Das Objekt mit der Beschriftung `Create Array` wird in einer offenen Ansicht dargestellt. Die offene Ansicht ist deutlich größer und ausführlicher. Eine ausführliche Beschreibung von Objektansichten enthält der Abschnitt “Ändern von Objektansichten” auf Seite 32 von Kapitel 1, “Verwenden der Agilent VEE-Entwicklungsumgebung.”

Erstellen von Bedienerschnittstellen in Agilent VEE

Ein weiterer Vorteil der Programmierung in VEE liegt darin, dass eine Bedienerschnittstelle in wenigen Minuten erstellt werden kann.

Mit dem Programm “Random” aus Abbildung I-2 werden die Objekte, die der Bediener sehen muss, ausgewählt und in einer **Fensteransicht** ("panel view") abgelegt. Eine Fensteransicht zeigt nur die Objekte an, die der Bediener zur Ausführung des Programms und zur Anzeige der resultierenden Daten benötigt. Abbildung I-3 zeigt die Fensteransicht des Programms “Random” in Abbildung I-2.

Hinweis

Das Programm und seine Bedienerschnittstelle sind verschiedene Ansichten des gleichen VEE-Programms. Sie können von einer Ansicht zur anderen wechseln, indem Sie die Schaltflächen "Detailansicht" und "Fensteransicht" in der Titelleiste des Fensters in VEE anklicken. Alle Änderungen oder Aktualisierungen, die Sie an einem Programm (Detailansicht) vornehmen, werden automatisch auch in der Bedienerschnittstelle (Fensteransicht) durchgeführt. Sie können die Bedienerschnittstelle auch gegen unerwünschte Änderungen schützen.

Weitere Informationen zum Erstellen einer Bedienerschnittstelle finden Sie unter “Erstellen einer Fensteransicht (Bedienerschnittstelle)” auf Seite 95.

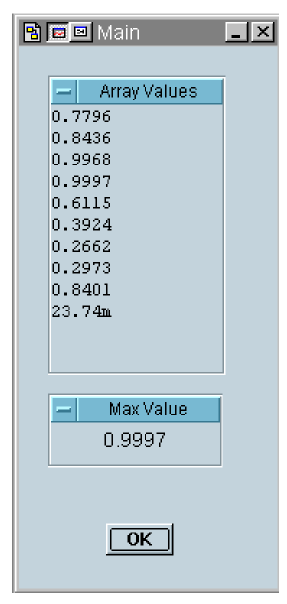


Abbildung I-3. Fensteransicht (oder Bedienerschnittstelle) des VEE-Programms

Mit VEE können Sie manche Aufgaben in Minutenschnelle ausführen, für die Sie in einer Textsprache Tage brauchen würden.

- Erstellen farbenprächtiger, intuitiver Oberflächen für Programme.
- Erstellen von Bedienerschnittstellen, die mit Tastatur und Maus oder nur über die Tastatur bedient werden.
- Auswählen einer breiten Palette von Funktionen für die Benutzereingabe und die Anzeige von Daten.
- Verwenden von Einblendfenstern zum Erstellen eines Fokus und zum Einsparen von Platz auf dem Bildschirm.
- Sichern von Programmen gegen unerwünschte Änderungen.

- Verwenden von Beschriftungen, Auswählen von Farben und Schriften sowie Hinzufügen von akustischen Signalen, Notizblocks, Schaltflächen und Schaltern in unterschiedlichsten Formaten.
- Verwenden eigener oder standardisierter ActiveX-Steuerelemente (nur PC) für die Benutzereingabe oder zum Anzeigen von Daten.

Nutzen vorhandener Testprogramme mit Agilent VEE

Für alle unterstützten Betriebssysteme bietet VEE Mechanismen zum Verbinden ("Link") konventioneller Testprogramme sowie kommerzieller Anwendungen. Sie können mit VEE beispielsweise vorhandene Tests in Rocky Mountain Basic, C, C++, Visual Basic, Fortran oder Pascal (oder jeder anderen kompilierten oder interpretierten Sprache auf Ihrem Betriebssystem) verwenden. VEE bietet auch eine Reihe von Funktionen zur prozessübergreifenden Kommunikation, um Daten mit kommerziellen Anwendungen wie Datenbanken oder Tabellenkalkulationen gemeinsam zu nutzen.

Auf PCs unterstützt VEE Standardverbindungen zur ActiveX-Automatisierung und zu Steuerelementen sowie DLLs. Unter HP-UX unterstützt VEE benannte Pipes und gemeinsame Bibliotheken.

Steuern von Instrumenten mit Agilent VEE

VEE bietet eine Reihe von Funktionen zum Steuern von und zur Kommunikation mit Instrumenten.

- Verwendung von Panel-Treibern (Instrumententreibern) für mehr als 450 Geräte verschiedener Hersteller sowie aller von verschiedenen Herstellern verfügbaren Treibern, sofern sie VXI*Plug&Play*-kompatibel mit den Rahmendefinitionen von Windows 95, Windows 98, Windows 2000, Windows NT 4.0 oder HP-UX sind.
- Verwendung der direkten E/A von VEE zum Senden von Befehlszeichenfolgen an Instrumente über Standardschnittstellen wie GPIB (IEEE - 488), GPIO, RS 232, VXI oder LAN-orientierten Instrumenten für ferne Tests.

- Steuern von PC-Zusatzkarten beliebiger Hersteller, sofern die Karten einen standardisierten ODAS-Treiber oder eine Dynamic Link Library unterstützt.
- Verwendung einer direkten VXI-Backplane-Steuerung mit integrierten PCs oder Workstations.
- Steuern einer Vielzahl von Gerätetypen mit einfachen, organisierten Funktionen zur Instrumentenverwaltung.

Verbessern der Testmöglichkeiten mit Agilent VEE

Die VEE-Produkte bieten folgende Funktionen und Vorteile:

- Weniger Zeitaufwand für Entwicklung und Wartung durch grafische Programmierung.
- Integration mit konventionellen Sprachen wie C, C++, Visual Basic, Pascal, Fortran und RMB.
- Komfortable und flexible Funktionen der Bedienerschnittstelle.
- Unterstützung der meisten gängigen Testplattformen.
- Verwendung von ActiveX-Automatisierung und -Steuerelementen.
- Hervorragende Support-Optionen von Agilent Technologies.
- Bedienerfreundliche und leistungsstarke Dokumentationshilfen.
- Einfache Portierung von Testdaten in Standard-Kalkulationstabellen und Textverarbeitungsprogramme für Berichte.
- Tools zur prozessübergreifenden Kommunikation zur Verbindung mit anderen Anwendungen wie relationalen Datenbanken oder Paketen zur Statistikanalyse (nur VEE Pro 6.0).
- Fehlerbehebungs-Tools zur effizienteren Gestaltung von Entwicklung und Wartung großer, komplexer Programme (nur VEE Pro 6.0).

- Leistungsstarke Testausführungs-Tools im Produkt enthalten (nur VEE Pro 6.0).
- Möglichkeiten zu fernen Tests mit den Web-Monitorfunktionen von VEE (nur VEE Pro 6.0).
- Unbegrenzte Laufzeit für die Verteilung Ihrer Programme (nur VEE Pro 6.0).
- Preisgünstige Standortlizenzen (nur VEE Pro 6.0).

Installieren von Agilent VEE und Informationen zum Produkt

Dieser Abschnitt enthält Anleitungen zur Installation und Verwendung von VEE: Installieren von VEE, Informationen zu VEE, Verwenden von VEE und Anfordern von VEE-Support.

Installieren von Agilent VEE und den E/A-Bibliotheken

Informationen zur Installation von VEE Pro 6.0 und E/A-Bibliotheken finden Sie in den Installationsunterlagen, die Sie zusammen mit VEE erhalten haben. (Die E/A-Bibliotheken werden von VEE zur Kommunikation mit Instrumenten verwendet.)

Installing and Distributing VEE Pro 6.0 RunTime (in der Online-Hilfe) beschreibt die Installation und Verteilung der Laufzeitversion ("RunTime version") von VEE Pro 6.0. Die Laufzeitversion wird zur Ausführung von VEE-Programmen auf PCs verwendet, auf der keine VEE-Software installiert ist. Weitere Informationen zur Laufzeitumgebung finden Sie in der Online-Hilfe. Wählen Sie `Help`, `Contents and Index` und `Installing and Distributing Agilent VEE Pro RunTime` aus. Sie können diese Informationen auch drucken.

Informationen zu Agilent VEE

Für Informationen zu VEE können Sie die VEE-Multimedia-Lernprogramme ansehen, die Online-Hilfe aufrufen, in den Handbüchern (beispielsweise dem vorliegenden) nachschlagen und VEE-Seminare besuchen.

- *VEE Multimedia-Lernprogramme:* Die VEE-Multimedia-Lernprogramme im Menü `Help` ⇒ `Welcome` von VEE sind Video-Präsentationen, in denen viele der Konzepte von VEE erläutert werden. Diese Präsentationen demonstrieren die Verwendung von VEE-Menüs, das Bearbeiten von Objekten und die Ausführung von Programmen. Jede Präsentation dauert ca. drei oder vier Minuten; Sie können diese Präsentationen beliebig oft anzeigen. Sie können das Lernprogramm unterbrechen und VEE ausführen, um das Gelernte auszuprobieren, und anschließend das Lernprogramm fortsetzen.

- *VEE Online-Hilfe:* Eine Möglichkeit, die neuen Funktionen von VEE kennen zu lernen, ist die Auswahl von `Help` ⇒ `Contents and Index` ⇒ `What's New in Agilent VEE 6.0`. Unter `Help` ⇒ `Welcome` ⇒ `Introduction` finden Sie einen Überblick über das VEE-Produkt.

  Die Online-Hilfe umfasst eine Reihe weiterer Funktionen. Weitere Informationen hierzu finden Sie unter “Hilfe aufrufen” auf Seite 26 und “Verwenden der Online-Hilfe” auf Seite 103.

- *VEE Handbücher:* Das Handbuchpaket für VEE umfasst dieses *VEE Pro Benutzerhandbuch* und das Handbuch *VEE Pro Advanced Techniques*.

- *Agilent VEE Seminare:* Informationen zu VEE-Seminaren finden Sie in der Web-Site `http://www.agilent.com/comms/education`.

Hinweis

Die VEE-Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter `Help` ⇒ `Open Example...` ⇒ `Manual` ⇒ `UsersGuide` enthalten.

Anfordern einer kostenlosen Testversion der Software

Eine kostenlose Testversion der Software ist auf CD verfügbar oder kann von der VEE-Web-Site heruntergeladen werden. Zum Anfordern der Agilent Technologies VEE Evaluation Kit CD rufen Sie unter +49 1805-246333 an oder wenden Sie sich an sich an Ihre lokale Agilent Technologies-Niederlassung. Eine Übersicht über die weltweiten Niederlassungen finden Sie unter:

`http://www.agilent.com`

MATLAB Script - Überblick

MATLAB® Script ist eine Untermenge des Standard-Vollprodukts MATLAB von The MathWorks. Dieses Produkt gibt dem Benutzer direkten Zugriff auf die Kernfunktionen von MATLAB, wie beispielsweise höhere Mathematik, Datenanalyse sowie wissenschaftliche und Engineering-Grafik. Das Objekt MATLAB kann sehr einfach in ein beliebiges Agilent VEE-Programm einbezogen werden.

MATLAB Script umfasst Hunderte von Funktionen für:

- Datenanalyse und -darstellung
- Numerische Berechnung einschließlich:
  - Lineare Algebra und Matrixberechnung
  - Fourier- und statistische Analyse
  - Lösen von Differentialgleichungen
  - Trigonometrische und fundamentale mathematische Operationen
- Engineering- und wissenschaftliche Grafik wie beispielsweise:
  - 2-D- und 3-D-Anzeige einschließlich Dreiecks- und Gitterdaten
  - Volumendarstellung von skalaren oder Vektordaten
  - Quiver-, Band-, Streu-, Balken-, Kreis- und Stamm-Plots

Toolbox zur Signalverarbeitung

MATLAB Script für VEE umfasst auch eine Untermenge der MATLAB Signalverarbeitungs-Toolbox, die auf einer soliden Basis von Techniken aus den Bereichen Filter-Design und Spektralanalyse aufbaut. Es stehen Funktionen zur Verfügung für:

- Signal- und lineare Systemmodelle
- Analoges Filter-Design
- FIR und IIR digitales Filter-Design, Analyse und Implementierung
- Transformationen wie FFT und DCT
- Spektralschätzung und statistische Signalverarbeitung
- Parametrische Zeitserien-Modellierung
- Waveform-Generierung

Informationen zum Vollprodukt MATLAB

MATLAB ist eine integrierte technische Computing-Umgebung, die numerische Berechnungen und erweiterte Grafik- und Darstellungsfunktionen mit einer höheren Programmiersprache verbindet. MATLAB umfasst Hunderte von Funktionen für:

- Datenanalyse und -darstellung
- Numerische und symbolische Berechnung
- Engineering- und wissenschaftliche Grafik
- Modellierung, Simulation und Prototyp-Verarbeitung
- Programmierung, Anwendungsentwicklung und GUI-Design

MATLAB wird in einer Vielzahl von Anwendungsbereichen eingesetzt, beispielsweise in der Signal- und Bildverarbeitung, im Design von Steuersystemen, im Financial Engineering und in der medizinischen Forschung. Die offene Architektur macht es sehr einfach, MATLAB und ergänzende Produkte bei der Untersuchung von Daten und der Erstellung spezifischer Tools einzusetzen, die zu einem sehr frühen Zeitpunkt wertvolle Informationen liefern und so einen echten Vorsprung im Wettbewerb bieten.

Als VEE-Benutzer können Sie die volle Leistung von MATLAB und der Signalverarbeitungs-Toolbox für Anwendungen mit Datenanalyse, Darstellung und Modellierung implementieren. Durch die Erweiterung auf die Vollversion dieser Produkte können Sie ein breites Spektrum zusätzlicher MATLAB-Funktionen in VEE-Anwendungen einsetzen, beispielsweise zum Erstellen von benutzerdefinierten Funktionen (M-Dateien), und haben Zugriff auf das MATLAB-Befehlsfenster, den MATLAB-Editor/Debugger und die grafische Benutzeroberfläche (GUI) der Signalverarbeitung.

Hinweis

Weitere Informationen zur Verwendung der MATLAB Script-Objekte in VEE-Programmen finden Sie in “Verwenden von MATLAB Script in Agilent VEE” auf Seite 194 von Kapitel 4, “Analysieren und Anzeigen von Testdaten.”

Anfordern von Support zu Agilent VEE

Sie können VEE-Support über das Web oder per Telefon (für Hilfestellung bei der Einrichtung) anfordern.

Abrufen von Informationen über das World Wide Web

Die VEE-Web-Site bietet vielfältige Informationen wie Anwendungshinweise, Benutzertipps, technische Daten und Informationen zu VEE-Partnern wie beispielsweise Herstellern von PC-Zusatzkarten.

- *Oberste VEE-Web-Site*: `http://www.agilent.com/find/tm_d_a_ch`.
- *Aktuelle Support-Informationen:* Während die Verbindung zum Netzwerk aktiv ist, klicken Sie in VEE die Option `Help` ⇒ `Agilent VEE on the Web` ⇒ `Support` an. Abbildung I-4 zeigt die Auswahl von Support in VEE. Sie können auch im Web-Browser `http://www.agilent.com/find/vee` auswählen und “Support” anklicken.

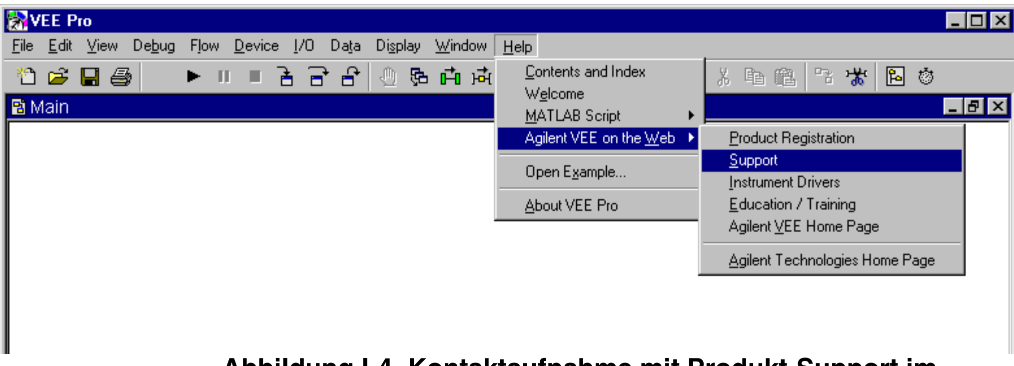


Abbildung I-4. Kontaktaufnahme mit Produkt-Support im VEE-Hilfemenü

- *Für ergänzende Hilfestellung bei der Einrichtung:* siehe "Phone Support Information" in der Online-Hilfe. Klicken Sie in VEE die Option `Help` ⇒ `Contents and Index` an und wählen Sie `Agilent VEE Support` aus.

Zusätzliche Informationsquellen zu MATLAB

Ausführliche Informationen zur Verwendung des Objekts MATLAB Script erhalten Sie über den MATLAB Script Help Desk. Wählen Sie in VEE die Option `Help` ⇒ `MATLAB Script` ⇒ `Help Desk` aus. Der `MATLAB Help Desk` wird in einem Web-Browser angezeigt.

Weitere Informationen zu MATLAB, MATLAB-Toolboxen und anderen Produkten von The MathWorks finden Sie im World Wide Web unter `www.mathworks.com`, oder rufen Sie an: +001-508-647-7000.

Weitere Informationsquellen sind:

- Vollständige MATLAB-Dokumentation: `www.mathworks.com/access/helpdesk/help/helpdesk.shtml`
- MATLAB-Aktualisierungsangebot: Ein spezielles Angebot ist verfügbar für Benutzer von VEE Pro 6.0 und VEE OneLab 6.0. Weitere Informationen finden Sie unter `www.mathworks.com/veeupgrade`
- MATLAB-Produktinformationen: `www.mathworks.com/products`
- MATLAB Technische Hilfestellung: `www.mathworks.com/support`
- MathWorks Store: `www.mathworks.com/store`
- MathWorks Homepage: `www.mathworks.com`
- Usenet Newsgroup: Die Newsgroup `comp.soft-sys.matlab` bietet ein Forum für Experten und Anfänger, die mit MATLAB arbeiten und Fragen oder Kommentare zu MATLAB und den zugehörigen Produkten haben.

1

Verwenden der Agilent VEE-Entwicklungsumgebung

Verwenden der Agilent VEE-Entwicklungsumgebung

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Unterstützte Systeme
- Verwendung des Hilfesystems
- Starten von VEE
- Das VEE-Fenster
- Arbeiten mit Objekten
- Verwalten des Arbeitsbereichs
- Auswählen von Menüelementen
- Pins und Anschlüsse an VEE-Objekten
- Verbinden von Objekten zum Erstellen von Programmen
- Erstellen, Ausführen, Drucken, Speichern und Öffnen von Programmen
- Die Arbeitsweise von VEE-Programmen

Erforderliche Zeit für dieses Kapitel: 1,5 Stunden

Überblick

In diesem Kapitel lernen Sie, wie Sie VEE starten, Menüs verwenden und mit Objekten arbeiten. Sie lernen den Zweck von Pins und Anschlüssen in VEE kennen. Sie verbinden Objekte miteinander zum Erstellen eines einfachen VEE-Programms und lernen die Arbeitsweise von VEE-Programmen kennen.

Interaktion mit Agilent VEE

In diesem Abschnitt wird die Verwendung der grafischen Programmiersprache VEE erläutert. Sie finden hier auch eine Liste der unterstützten Systeme, Informationen zur Arbeitsweise der Maus und der Menüs sowie zum Aufrufen der Hilfe, zum Starten von VEE und zum Arbeiten im VEE-Fenster.

Unterstützte Systeme

Diese Version von VEE, Version 6.0, wird mit folgenden Systemen unterstützt:

- Windows 95, Windows 98, Windows 2000 und Windows NT 4.0 auf einem PC.
- HP-UX Workstations (Version 10.20 auf Serie 700). Diese Version von VEE kann nicht mit HP-UX Version 11.x oder älteren Versionen als 10.2 ausgeführt werden.

Die Maus und die Menüs

Wahrscheinlich haben Sie bereits Erfahrungen mit der maus- und menügesteuerten Benutzeroberfläche des Computers gesammelt und kennen Pull-down-Menüs, Symbolleisten und Dialogfenster, die Sie über die Maus oder die Tastatur steuern können. VEE verwendet die Oberfläche des Computers. In den Anleitungen zur Verwendung der Maus beim Arbeiten mit Menüs, Symbolen, Schaltflächen und Objekten werden die folgenden allgemeinen Techniken verwendet:

- Zum “Anklicken” eines Elements positionieren Sie den Mauszeiger auf das gewünschte Element; jetzt drücken Sie die *linke* Maustaste und lassen Sie gleich wieder los.
- Zum “Doppelklicken" auf ein Element positionieren Sie den Mauszeiger auf das gewünschte Element und klicken anschließend zwei Mal in schneller Folge mit der *linken* Maustaste.

- Zum “Ziehen” eines Elements positionieren Sie den Mauszeiger auf das gewünschte Elemente, *halten Sie die linke Maustaste gedrückt,* und verschieben Sie das Element an die gewünschte Position. Lassen Sie anschließend die Maustaste wieder los.

Hinweis

Die rechte Maustaste wird seltener verwendet. Wenn Sie für eine Aktion mit der rechten Maustaste klicken müssen, wird speziell darauf hingewiesen. Ihre Maus hat eventuell auch eine mittlere Taste - Sie verwenden diese Taste in VEE jedoch nicht.

Starten von Agilent VEE

Windows Klicken Sie `Start` ⇒ `Programs` ⇒ `Agilent VEE Pro 6.0` an.

HP-UX Geben Sie bei einer Shell-Eingabeaufforderung in einem HP VUE- oder X11-Fenster folgendes ein: `veetest` **Eingabetaste**. (Die Variable `PATH` muss die Angabe `/usr/bin` enthalten.)

Das Agilent VEE-Fenster

Nach dem Installieren und Starten von VEE wird das in Abbildung 1-1 dargestellte VEE-Fenster angezeigt.

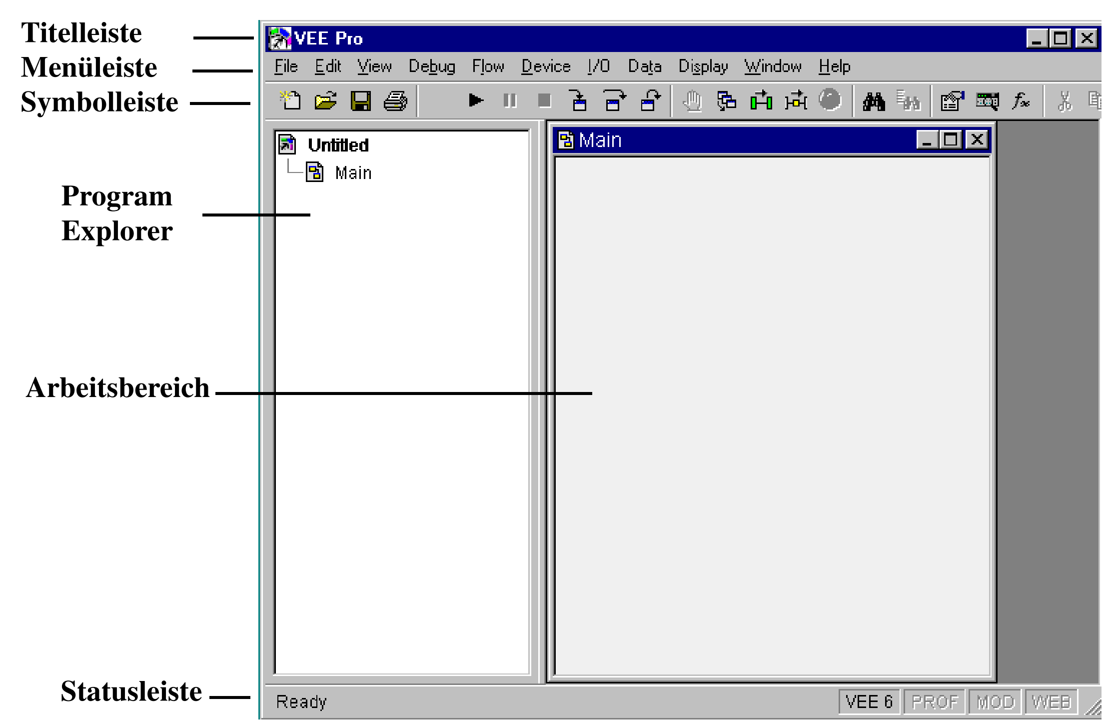


Abbildung 1-1. Die VEE-Entwicklungsumgebung

Diese Elemente beschreiben die Komponenten des VEE-Fensters.

Titelleiste Die oberste Zeile im Fenster enthält das VEE-Symbol, den Fensternamen, die Schaltflächen zum Anzeigen des Fensters in Symbol- oder Vollbildgröße sowie die Schaltfläche zum Schließen. Sie verschieben das Fenster durch Ziehen der Titelleiste. Klicken Sie das VEE-Symbol an, um das Fenstermenü aufzurufen.

Menüleiste Die zweite Zeile enthält die Menüelemente mit den verschiedenen VEE-Befehlen und -Objekten.

Symbolleiste Die dritte Zeile enthält Symbole oder Schaltflächen für den direkten Zugriff (“Direktaufrufe”) auf die am häufigsten verwendeten Menübefehle. (Platzieren Sie den Mauszeiger auf einer Schaltfläche - VEE zeigt die jeweilige Funktion an.)

Arbeitsbereich Ein Bereich im Programmfenster (Bearbeitungsfenster) wie beispielsweise `Main`, `UserObject` oder `UserFunction`, in denen Sie Objekte positionieren und miteinander verbinden.

Program Explorer Ein Bereich auf der linken Seite des VEE-Fensters, der die Struktur des VEE-Programms zeigt. Die obere Ecke zeigt den Namen des aktuellen Programms, z. B. `myprog.vee`, oder sie zeigt den Hinweis `Untitled` ("Ohne Namen") an.

Der Program Explorer ermöglicht das Wechseln zwischen den Programmfenstern. Zum Ändern der Größe des Program Explorer verschieben Sie den normalen Zeiger auf die rechte Begrenzung, bis er als vertikaler Teiler angezeigt wird, klicken Sie mit der Maustaste, und verschieben Sie den Zeiger an die gewünschte Position.

Haupt-fenster	Ein Fenster mit einem Arbeitsbereich, in dem Sie VEE-Programme entwickeln und bearbeiten. Es können auch weitere Programm-/Bearbeitungsfenster vorhanden sein, z. B. `UserObject`.
Status-leiste	Die untere Zeile zeigt Meldungen zum VEE-Status an. Sie enthält vier Statusanzeigen in der rechten Ecke. Die Anzeigen (von links nach rechts) enthalten folgende Informationen: Den Ausführungsmodus Den Status des Profilierers MOD wird angezeigt, wenn das Programm geändert wurde Der Web-Server ist aktiviert

Hinweis

Der Schwerpunkt dieses Handbuchs liegt auf VEE Version 6.0. Falls Sie eine ältere Version von VEE verwenden (klicken Sie zum Überprüfen der Version `Help ⇒ About VEE Pro` an), stehen preisgünstige Aktualisierungen zur Verfügung. Wenn Sie eine Support-Vereinbarung für Software-Aktualisierungen haben, erhalten Sie die neue Version automatisch.

Hilfe aufrufen

VEE bietet ein Online-Hilfesystem (`Help`) für die VEE-Umgebung sowie Online-Hilfetexte zu einzelnen Objekten und Themen. Darüber hinaus können Sie weitere Informationen in der Dokumentation nachschlagen, die Sie mit dem Computer bzw. dem Betriebssystem erhalten haben. Die PC-Online-`Hilfe` enthält Informationen zu Themen wie beispielsweise:

- Auswahl von Befehlen in der Menüleiste
- Auswählen und Inaktivieren von Menüelementen
- Verwenden der Symbolleisten
- Informationen zu Titel- und Statusleisten
- Anklicken von Symbolen und Schaltflächen
- Arbeiten mit Dialogfenstern
- Arbeiten mit verschiedenen Fenstertypen
- Verwenden der Online-Hilfe

Beginnen Sie zunächst mit der Anzeige `Help`⇒ `Welcome`, über die Sie die VEE-Multimedia-Lernprogramme aufrufen können. Diese Begrüßungsanzeige ist in Abbildung 1-2 dargestellt.

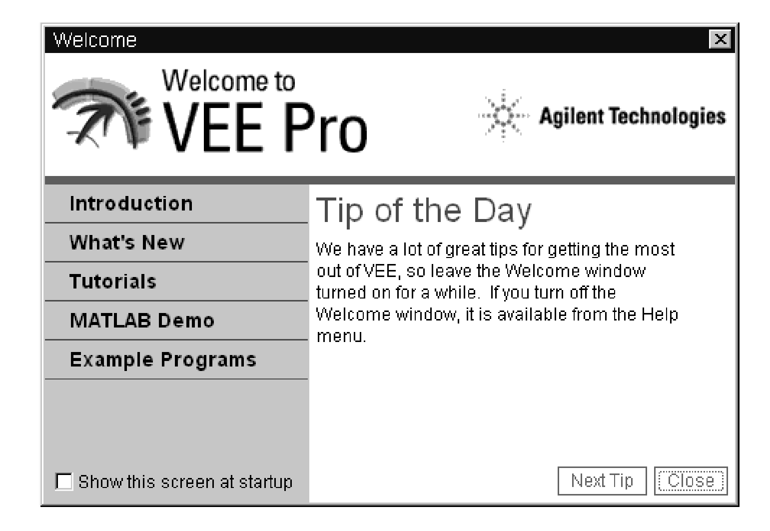


Abbildung 1-2. Die VEE-Begrüßungsanzeige in der Hilfe

Die VEE-Online-Hilfe wurde für Ihr Betriebssystem konzipiert. Klicken Sie `Help` an, um das in Abbildung 1-3 dargestellte Menü aufzurufen. Der Hilfetest umfasst ein Inhaltsverzeichnis und einen Index, die Begrüßungsanzeige (in dem die Lernprogramme aufgeführt sind), Gerätetreiber, Web-Site-Informationen, Beispiele sowie die Versionsnummer.

Zur Durchführung dieses Selbstlernkurses werden Sie die VEE-Dokumentation nicht benötigen; Sie finden in der Produktdokumentation jedoch ausführliche Informationen zu einzelnen Funktionen oder Konzepten. Verwenden Sie das Hilfesystem, um nach VEE-Themen zu suchen, über die Sie mehr erfahren wollen. Mit dem Hilfesystem können Sie zu verwandten Themen "springen".

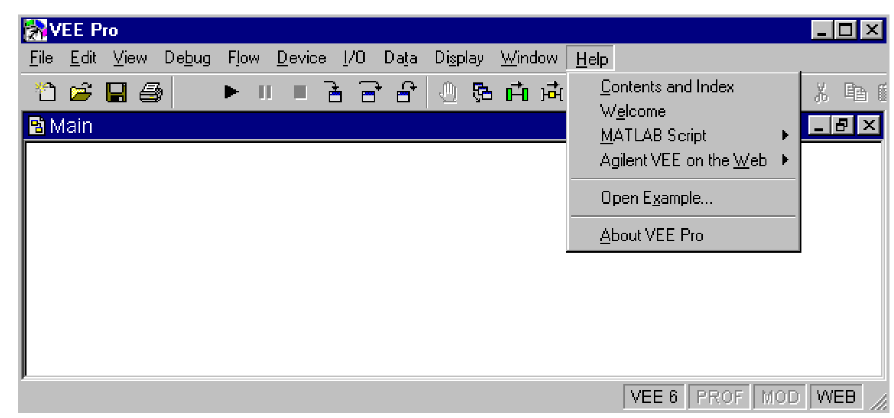


Abbildung 1-3. Verwendung des Hilfemenüs

Wählen Sie `Contents and Index` aus, um die VEE-Hilfe wie in Abbildung 1-4 dargestellt zu starten.

(Die Anzeigen unter HP-UX weisen gewisse Unterschiede auf.)

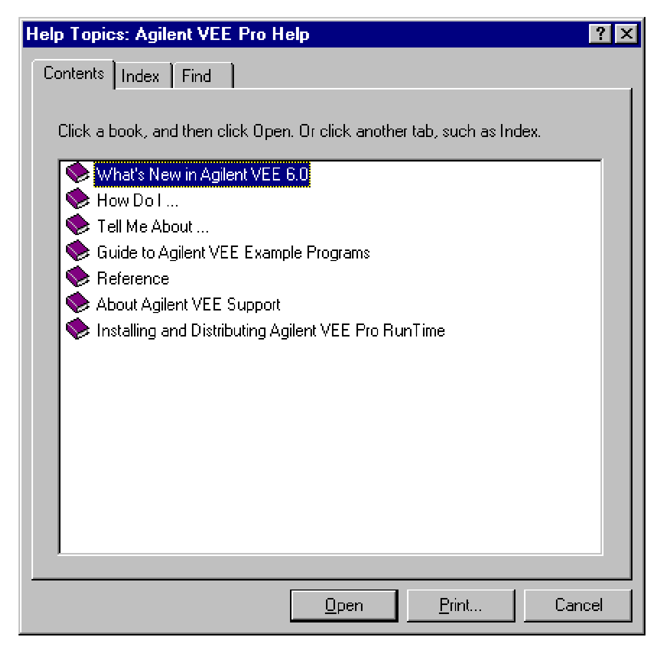


Abbildung 1-4. VEE Hilfe-Inhaltsverzeichnis

Das Register `Help Contents` (Inhaltsverzeichnis) enthält folgende Themen.

What's New in Agilent VEE 6.0	Erläutert neue Funktionen.
How Do I...	Enthält Informationen zur Vorgehensweise bei gängigen Aufgaben.
Tell Me About...	Erläutert die Konzepte von VEE.
Guide to Agilent VEE Example Programs	Zusammenfassung der mit VEE bereitgestellten Beispielprogrammen .
Reference	Enthält Referenzinformationen zu allen Funktionen und Objekten.
About Agilent VEE Support	Enthält Informationen zum Abrufen von Support-Leistungen zu VEE.
Installing and Distributing Agilent VEE Pro RunTime	Erläutert die Verteilung der VEE Pro RunTime-Umgebung.

Hinweis

Als Direktaufruf zum Abrufen von Hilfeinformationen zu einem bestimmten Objekt oder einem Dialogfenster drücken Sie **F1** auf der Tastatur. Sie können auch `Help` in einem Objektmenü anklicken, um spezifische Informationen zu dem jeweiligen Objekt abzurufen.

Weitere Informationen zu spezifischen Hilfefunktionen bei der Entwicklung von Programmen finden Sie im Abschnitt “Verwenden der Hilfefunktion” auf Seite 103.

Arbeiten mit Objekten

Ein VEE-Programm besteht aus verbundenen Objekten. Wählen Sie zum Erstellen eines Programms *Objekte* in VEE-Menüs aus, beispielsweise `Flow`, `Data` und `Display`. Verbinden Sie die Objekte über Linien, die an die Objekt-Pins "angeschlossen" werden. (Weitere Informationen zu Pins finden Sie im Abschnitt "Informationen zu Pins und Anschlüssen" auf Seite 49.) Mit einer Gruppe verbundener Objekte erstellen Sie ein Programm.

In diesem Abschnitt wird beschrieben, wie Sie Objekte in einem Programm auswählen und verwenden.

1. Starten Sie VEE. Klicken Sie `Start` ⇒ `Programs` ⇒ `Agilent VEE Pro 6.0` in Windows an, oder geben Sie `veetest` ein und drücken Sie die **Eingabetaste** bei einer Shell-Eingabeaufforderung in einem HP VUE- oder X11-Fenster in HP-UX (die Variable PATH muss die Angabe `/usr/bin` enthalten).

2. Verwenden Sie die Anleitungen in diesem Abschnitt zum Experimentieren mit Objekten.

Hinweis

Bei den späteren Übungen wird davon ausgegangen, dass Sie die VEE-Software gestartet haben. Wenn Sie Informationen zum Starten von VEE benötigen, schlagen Sie auf dieser Seite oder im Abschnitt "Starten von Agilent VEE" auf Seite 24 nach.

Hinzufügen von Objekten zum Arbeitsbereich

Öffnen Sie ein entsprechendes Menü, klicken Sie das gewünschte Objekt an, ziehen Sie das Objekt an die Zielposition im Arbeitsbereich und klicken Sie mit der Maustaste (der Umriss verschwindet, und das Objekt wird angezeigt).

1. Wenn Sie beispielsweise ein Objekt `Function Generator` dem Arbeitsbereich hinzufügen wollen, wählen Sie `Device` ⇒ `Virtual Source` ⇒ `Function Generator` in der Menüleiste aus, wie in Abbildung 1-5 gezeigt.

Hinweis

Der Pfeil rechts von `Virtual Source` kennzeichnet ein Untermenü. Drei Punkte hinter einem Menüelement weisen darauf hin, dass ein oder mehrere Dialogfenster folgen. `File` ⇒ `Save As...` ist ein Beispiel für diese Arbeitsweise.

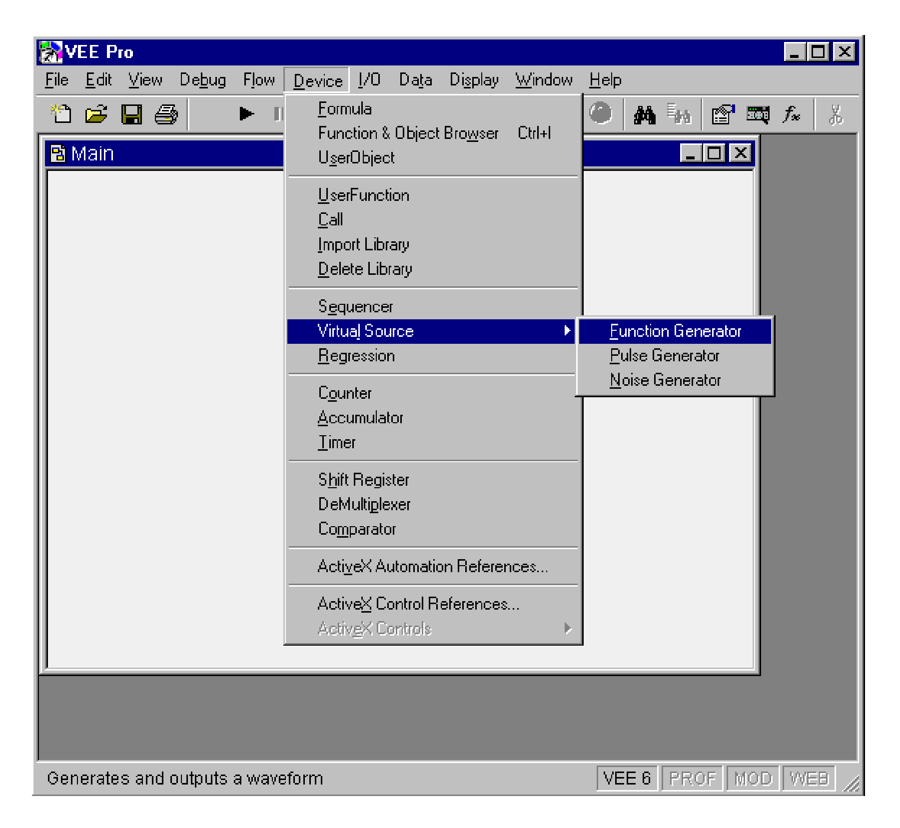


Abbildung 1-5. Hinzufügen von Objekten zum Arbeitsbereich

Ein Umriss des Objekts erscheint im Arbeitsbereich.

2. Verschieben Sie den `Function Generator` in die Mitte des Arbeitsbereichs und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Der `Function Generator` erscheint wie in Abbildung 1-6 dargestellt.

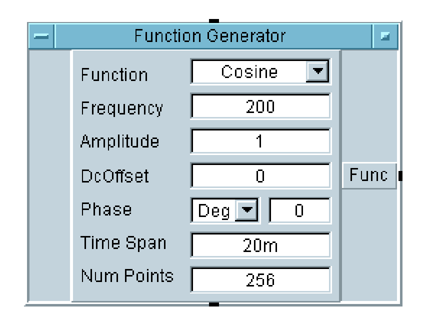


Abbildung 1-6. Hinzufügen eines Objekts "Function Generator"

Nachdem Sie ein Objekt im Arbeitsbereich platziert haben, können Sie es durch Ziehen seiner Titelleiste verschieben, genau wie beim Verschieben eines Fensters.

Hinweis

Im weiteren Verlauf dieses Handbuchs wird für die Anleitungen eine "Kurzschrift" verwendet. Das Auswählen des Objekts `Function Generator` wird beispielsweise im folgenden Format angegeben:

`Device` ⇒ `Virtual Source` ⇒ `Function Generator`

Hinweis

Sie können auf dem Bildschirm mehr Platz schaffen, indem Sie `View` ⇒ `Program Explorer` anklicken. Die Auswahl des Program Explorer wird dadurch aufgehoben, und er wird vom Bildschirm entfernt. Menüelemente sind “ausgewählt”, wenn davor ein Häkchen angezeigt wird.

Ändern von Objektansichten

VEE zeigt Objekte in der “Symbolansicht” oder der “offenen Ansicht” an, wie in Abbildung 1-7 dargestellt.

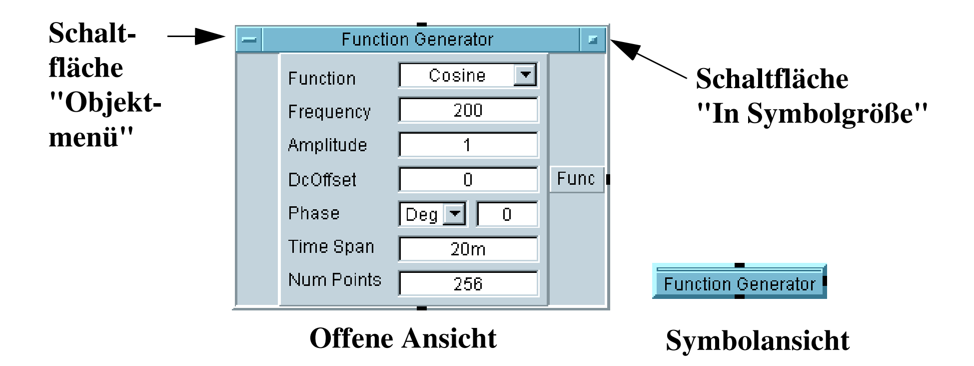


Abbildung 1-7. Objekt in offener Ansicht und Symbolansicht

Die Symbolansicht spart Platz im Arbeitsbereich und macht Programme besser lesbar. Die offene Ansicht bietet mehr Detailinformationen und ermöglicht das Bearbeiten der Eigenschaften und Einstellungen eines Objekts.

1. Klicken Sie zum Wechseln von der offenen zur Symbolansicht die Schaltfläche **Symbolgröße** (das Feld am rechten Rand der Titelleiste des Objekts) an.

2. Klicken Sie zur Rückkehr in die offene Ansicht das Objekt Symbolansicht (an einer beliebigen Stelle des Objekts) doppelt an.

Hinweis

Das Objektmenü enthält ebenfalls Auswahloptionen für die Symbolgröße (`Minimize`) und zum Wiederherstellen (`Restore`). Zum Anzeigen der Objektmenüs klicken Sie die Schaltfläche **Objektmenü** am linken Rand der Titelleiste an, oder klicken Sie mit der rechten Maustaste eine beliebige Stelle des Objekts an.

Nicht alle Objekte haben die gleiche Struktur oder die gleichen Elemente; für alle Objekte gilt jedoch: Sie können das Objekt in der offenen Ansicht bearbeiten oder es in der Symbolansicht anzeigen, um Platz zu sparen.

Auswählen eines Objektmenüs

Jedes VEE-Objekt hat ein **Objektmenü**, über das Aktionen mit dem Objekt ausgeführt werden können, z. B. `Clone`, `Size`, `Cut`, `Move` und `Minimize`. Die meisten Objekte haben ähnliche Attribute, es gibt jedoch je nach der Funktionalität des Objekts gewisse Unterschiede. Ausführliche Informationen zum Objektmenü eines bestimmten Objekts finden Sie in der Online-Hilfe.

1. Klicken Sie zur Auswahl des Objektmenüs *ein Mal* die Schaltfläche "Objektmenü" an. (Alle Objektmenüs werden auf die gleiche Weise geöffnet.) Das Objektmenü erscheint wie in Abbildung 1-8 dargestellt. (Klicken Sie die Schaltfläche "Objektmenü" nicht doppelt an. Das doppelt Anklicken des Objektmenüs ist der Direktaufruf zum Löschen des Objekts.)

2. Sie können jetzt die Optionen des Objektmenüs anklicken, um die gewünschte Aktion auszuführen. Zum Verlassen des Menüs klicken Sie einfach einen leeren Bereich *außerhalb* des Menüs an.

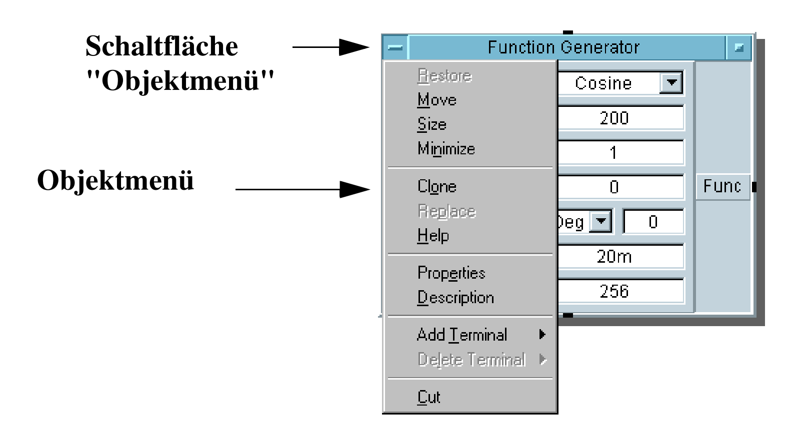


Abbildung 1-8. Auswählen eines Objektmenüs

Direktaufruf: Sie können das Objektmenü auch auswählen, indem Sie den Mauszeiger auf eine beliebige Stelle auf dem Objektkörper positionieren und mit der *rechten* Maustaste klicken. Dies funktioniert für die offene und die Symbolansicht.

Verschieben eines Objekts

1. Zum Verschieben des Objekts `Function Generator` klicken Sie `Move` im Objektmenü an; klicken Sie anschließend mit der linken Maustaste und halten Sie diese Taste gedrückt. Daraufhin wird ein Umriss des Objekts angezeigt.

2. Verschieben Sie den Umriss an die neue Position, und halten Sie die Maustaste weiterhin gedrückt, wie in Abbildung 1-9 dargestellt. Lassen Sie die Maustaste los, um die Objekte an die neue Position zu verschieben.

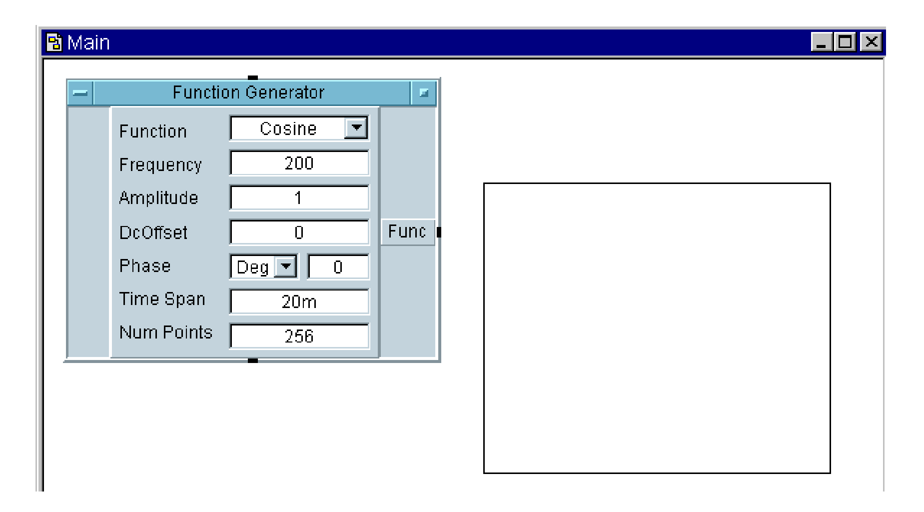


Abbildung 1-9. Verschieben eines Objekts

Sie können Objekte auch wie folgt verschieben:

- Klicken Sie den Titelbereich der offenen Ansicht eines Objekts an, und ziehen Sie das Objekt an eine neue Position.
- Klicken Sie einen beliebigen Teil eines Objekts in der offenen Ansicht (*außer* Schaltflächen, Eingabefeldern, Pins, Anschlüssen oder den vier Ecken zum Ändern der Objektgröße) an, und ziehen Sie das Objekt an die neue Position.
- Klicken Sie einen beliebigen Teil des Objekts in der Symbolansicht *außer* in der Nähe der vier Ecken zum Ändern der Objektgröße an, und ziehen Sie das Objekt an die neue Position.

Hinweis

"Object Location Information" in der Statusleiste (am unteren Rand des VEE-Fensters) zeigt die X- und Y-Position (in Pixel) der linken oberen Ecke des Umrisses in Relation zur linken oberen Ecke des Arbeitsbereichs an. Zum Anzeigen der genauen Position eines Objekts klicken Sie mit der linken Maustaste ein Objekt an, um es auszuwählen, und halten Sie die linke Maustaste gedrückt. Die Position wird in der Statusleiste angezeigt. Verwenden Sie diese Informationen beim Platzieren eines Objekts an einer exakten Position.

Duplizieren (Klonen) eines Objekts

Die Operation **Clone** erstellt ein exaktes Duplikat eines Objekts einschließlich aller vorgenommenen Änderungen wie beispielsweise dem Umbenennen oder dem Ändern der Größe. Das Klonen ist eine Kurzform für das Ausschneiden und Einfügen.

1. Öffnen Sie das Objektmenü und wählen Sie `Clone` aus. Ein Umriss des duplizierten Objekts wird angezeigt.

2. Verschieben Sie den Umriss an die gewünschte Position und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Das geklonte Objekt erscheint, während das ursprüngliche Objekt weiterhin erhalten bleibt. In Abbildung 1-10 wurde der `Function Generator` bereits ein Mal geklont, und im Objektmenü ist der Befehl für ein erneutes Klonen ausgewählt.

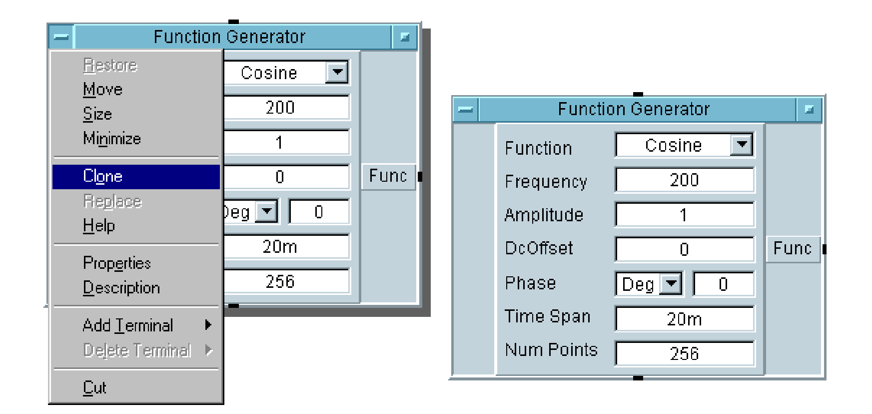


Abbildung 1-10. Klonen eines Objekts

Kopieren eines Objekts

Mit dieser Aktion wird ein Objekt in die Zwischenablage kopiert, sodass Sie es in VEE oder in einer anderen Anwendung wie MS Paint oder MS Word einfügen können.

1. Klicken Sie ein Objekt an, um es hervorzuheben, und klicken Sie anschließend `Edit` ⇒ `Copy` an.

   -ODER-

   Klicken Sie ein Objekt an, um es hervorzuheben, und drücken Sie anschließend **Strg-C**.

Löschen (Ausschneiden) eines Objekts

Zum Löschen (oder Ausschneiden) eines Objekts aus dem Arbeitsbereich öffnen Sie das Objektmenü des betreffenden Objekts und klicken `Cut` an. Wechseln Sie beispielsweise zum Objektmenü für den `Function Generator`, und klicken Sie `Cut` an. Das Objekt verschwindet aus dem Arbeitsbereich; es wird jedoch im **Ausschneidepuffer** gespeichert.

1. Öffnen Sie das Objektmenü und wählen Sie `Cut` aus.

   -ODER-

   Wählen Sie das Objekt aus (durch Anklicken) und drücken Sie **Strg-X**.

   -ODER-

   Platzieren Sie die Maus auf dem Objektmenü und klicken Sie doppelt.

Hinweis Gehen Sie hierbei vorsichtig vor, da ein Objekt durch *Doppelklicken* auf die Schaltfläche "Objektmenü" leicht versehentlich gelöscht wird. Wenn Sie ein Objekt versehentlich löschen, verwenden Sie die Schaltfläche `Einfügen` in der Symbolleiste (oder `Edit` ⇒ `Paste`), um das Objekt und alle seine Verbindungen wiederherzustellen.

Einfügen eines Objekts (“Rückgängig machen” des Ausschneidevorgangs)

Gehen Sie zum Einfügen eines kopierten oder gelöschten (ausgeschnittenen) Objekts in den Arbeitsbereich wie folgt vor:

1. Nachdem ein Objekt kopiert oder gelöscht wurde, klicken Sie `Edit` ⇒ `Paste` an. Daraufhin wird ein Umriss des Objekts angezeigt. Platzieren Sie das Objekt und klicken Sie mit der Maustaste, um es freizugeben.

   -ODER-

   Drücken Sie **Strg-V**.

Hinweis Wenn Linien an das Objekt angeschlossen waren, werden diese Verbindungen beibehalten. Diese Aktion funktioniert wie ein "Rückgängig machen" in anderen Programmen. Die Funktion heißt jedoch nicht "Rückgängig" oder "Undo", weil sie nicht für alle VEE-Programmaktionen verwendet werden kann. (Sie funktioniert auch mit gelöschten Objektgruppen.)

Ändern der Größe eines Objekts

1. Positionieren Sie den Mauszeiger auf einer der vier Ecken des Objekts, bis ein Pfeil für die Größenänderung angezeigt wird. Klicken-und-ziehen Sie jetzt das Objekt auf die gewünschte Größe. Lassen Sie die Maustaste los, um die neue Größe zu übernehmen. Abbildung 1-11 zeigt ein Objekt, dessen Größe mit dem Größenpfeil geändert wurde.

   -ODER-

   Öffnen Sie das Objektmenü und klicken Sie `Size` an. Der Mauszeiger wird als “Klammer rechts unten" angezeigt. Verschieben Sie die Klammer an die gewünschte Position der rechten unteren Ecke, und klicken Sie mit der Maustaste, um die neue Größe zu übernehmen.

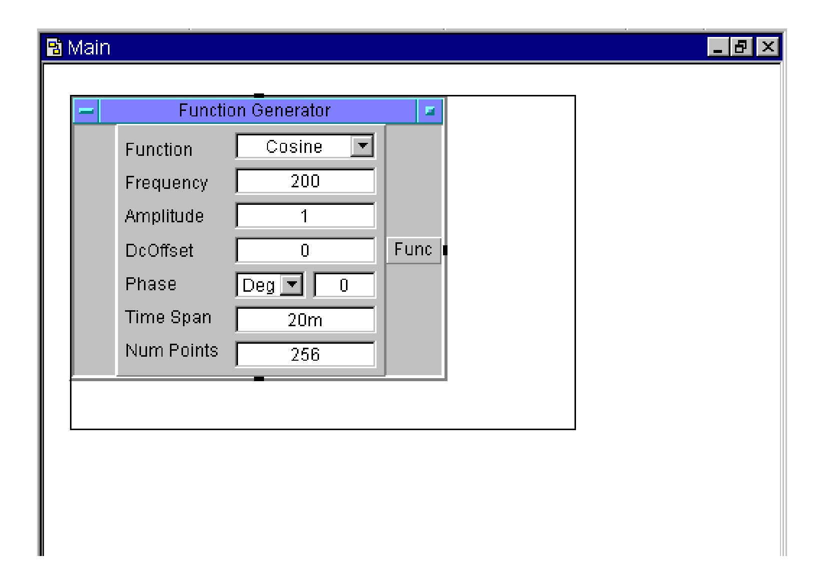


Abbildung 1-11. Ändern der Größe eines Objekts

Ändern des Namens (Titels) eines Objekts

1. Öffnen Sie das Objektmenü und wählen Sie `Properties...` aus. In dem daraufhin angezeigten Dialogfenster "Properties" (Eigenschaften) ist der aktuelle Titel hervorgehoben, wie in Abbildung 1-12 gezeigt.
2. Geben Sie den neuen Titel ein und klicken Sie `OK` an. Der neue Titel wird im Titelbereich angezeigt. Wenn Sie das Objekt in Symbolgröße anzeigen, erscheint der neue Titel in dem Symbol.

-ODER-

1. Klicken Sie doppelt auf die Titelleiste des Objekts, um das Dialogfenster "Properties" direkt aufzurufen.
2. Geben Sie den neuen Titel ein und klicken Sie `OK` an.

Hinweis

Mit der richtigen Anwendung von Tastatur- und Maustechniken können Sie bei der Bearbeitung viel Zeit sparen. Wenn Sie beispielsweise im Feld `Titel` im Dialogfenster `Properties` ganz am linken Rand des Bearbeitungsbereichs klicken, wird dort ein Cursor angezeigt. Sie können jetzt neuen Text eingeben, ohne den vorhandenen Text zu löschen.

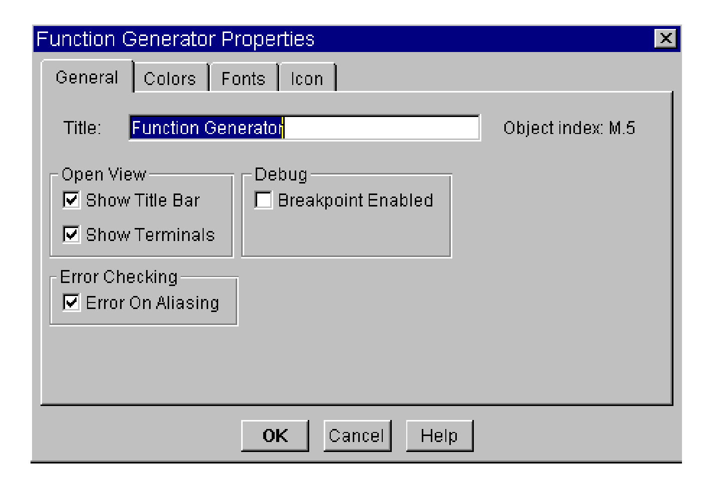


Abbildung 1-12. Ändern eines Objekttitels

Auswählen von Objekten oder Auswahl aufheben

1. Klicken Sie zum Auswählen eines Objekts das Objekt an. Hinter dem Objekt wird ein Schatten angezeigt. In Abbildung 1-13 ist beispielsweise das Objekt `For Count` ausgewählt.

2. Verschieben Sie zum Aufheben der Auswahl eines Objekts den Mauszeiger auf einen beliebigen offenen Bereich und klicken Sie mit der Maustaste. Der Schatten verschwindet. In Abbildung 1-13 ist beispielsweise das `Formula`-Objekt nicht ausgewählt.

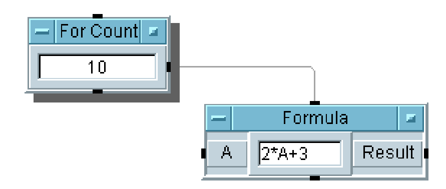


Abbildung 1-13. Ausgewählte und nicht ausgewählte Objekte

Hinweis

Das Wort "Auswählen" wird auch zur Auswahl eines Menüelements verwendet; durch den jeweiligen Kontext ist die Bedeutung jedoch klar.

Auswählen mehrerer Objekte

Beim Anklicken eines Objekts wird nur jeweils ein Objekt ausgewählt. Wenn Sie erneut klicken, um ein weiteres Objekt auszuwählen, wird die Auswahl des vorigen Objekts aufgehoben, und der entsprechende Schatten verschwindet. Wenn Sie mehrere Objekte auswählen wollen, um eine Operation (beispielsweise `Cut`) mit allen diesen Objekten auszuführen, gehen Sie wie folgt vor:

1. Drücken Sie die Taste **Strg** und halten Sie sie gedrückt, während Sie auf die einzelnen Objekte klicken. Lassen Sie die Taste **Strg** los, wenn alle gewünschten Objekte hervorgehoben werden.

   -ODER-

Drücken Sie **Strg**, und klicken-und-ziehen Sie ein Rechteck um die auszuwählenden Objekte. Die ausgewählten Objekte werden mit Schatten angezeigt.

Auswählen/Auswahl aufheben aller Objekte

1. Wenn Sie alle Objekte auswählen wollen, klicken Sie `Edit` ⇒ `Select All` an. (Oder drücken Sie **Strg-A**.)

2. Klicken Sie auf einen offenen Bereich im Fenster, um die Auswahl aller Objekte aufzuheben.

Kopieren mehrerer Objekte

1. Kopieren Sie die ausgewählten Objekte, indem Sie den Cursor auf einem Objekt platzieren. Drücken Sie die Taste **Strg** und halten Sie sie gedrückt, während Sie mit der linken Maustaste die Objekte (Umrisse) an die gewünschte Position ziehen. Ein neues Exemplar jedes Objekts wird an der gewünschten Position angezeigt.

   -ODER-

   Verwenden Sie `Edit` ⇒ `Copy`, um die ausgewählten Objekte in den Ausschneidepuffer zu kopieren. Klicken Sie `Paste` (im Menü `Edit` oder in der Symbolleiste) an, verschieben Sie die Objekte (Umrisse) an die gewünschte Position, und klicken Sie mit der linken Maustaste. Abbildung 1-14 zeigt die Objekte beim Kopieren an.

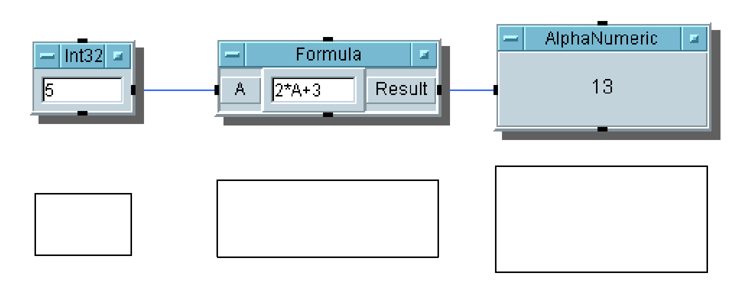


Abbildung 1-14. Mehrere Objekte beim Kopieren

Hinweis

In VEE für Windows werden Objekte, die Sie ausschneiden oder kopieren, auch in der Zwischenablage platziert. Sie können diese Objekte in anderen Windows-Anwendungen einfügen, sofern die jeweilige Anwendung die Windows-Zwischenablage unterstützt.

Bearbeiten von Objekten

Es gibt mehrere Möglichkeiten, Objekte in VEE zu bearbeiten. Verschiedene Bearbeitungsmenüs zeigen unterschiedliche Optionen an. Wählen Sie ein Bearbeitungsmenü oder -symbol wie folgt aus:

1. Klicken Sie `Edit` in der VEE-Menüleiste an, um das Menü `Edit` anzuzeigen, und wählen Sie die gewünschte Operation aus. Die Befehle im Menü `Edit` sind überall in VEE gleich.

   -ODER-

   Klicken Sie ein Symbol in der VEE-Symbolleiste an. Die VEE-Symbolleiste enthält Symbole für häufig verwendete Bearbeitungsbefehle wie `Cut` (Ausschneiden), `Copy` (Kopieren) und `Paste` (Einfügen).

   -ODER-

   Öffnen Sie das Objektmenü des Objekts, indem Sie es anklicken, und wählen Sie die gewünschte Operation aus. Objektmenüs enthalten spezifische Bearbeitungsoperationen für ein Objekt wie beispielsweise das Menü `Properties`, die im Hauptmenü `Edit` nicht vorhanden sind. Die Befehle im Objektmenü unterscheiden sich je nach dem Typ des Objekts. Vergleichen Sie beispielsweise die Objektmenüs der Objekte `Device` ⇒ `Formula` und `I/O` ⇒ `To` ⇒ `File`. Die beiden Menüs enthalten unterschiedliche Optionen speziell für das jeweilige Objekt.

   -ODER-

   Positionieren Sie den Mauszeiger auf eine beliebige *leere* Stelle im Arbeitsbereich, und klicken Sie mit der *rechten* Maustaste. Daraufhin wird ein Einblendmenü `Edit` (Bearbeiten) angezeigt.

Hinweis

Inaktive Menüelemente werden mit einem anderen Schatten angezeigt als aktive Elemente (sie sind “abgeblendet”). Die Operationen `Cut`, `Copy` und `Clone` im Menü `Edit` erscheinen beispielsweise mit einem anderen Schatten als aktive Menüelemente, bis ein Objekt im Arbeitsbereich hervorgehoben wird.

Erstellen von Datenlinien zwischen Objekten

1. Klicken Sie einen Datenausgangs-Pin eines Objekts an (oder knapp daneben) und anschließend den Dateneingangs-Pin eines anderen Objekts, wie in Abbildung 1-15 dargestellt. (Es erscheint eine Linie hinter dem Zeiger, während Sie von einem Pin zum anderen wechseln.)

2. Lassen Sie die Maustaste los. VEE zeichnet eine Linie hinter den beiden Objekten. Beachten Sie, dass VEE beim neu Positionieren der Objekte die Linie zwischen den Objekten beibehält.

Hinweis

Weitere Informationen zu Pins finden Sie unter “Informationen zu Pins und Anschlüssen” auf Seite 49.

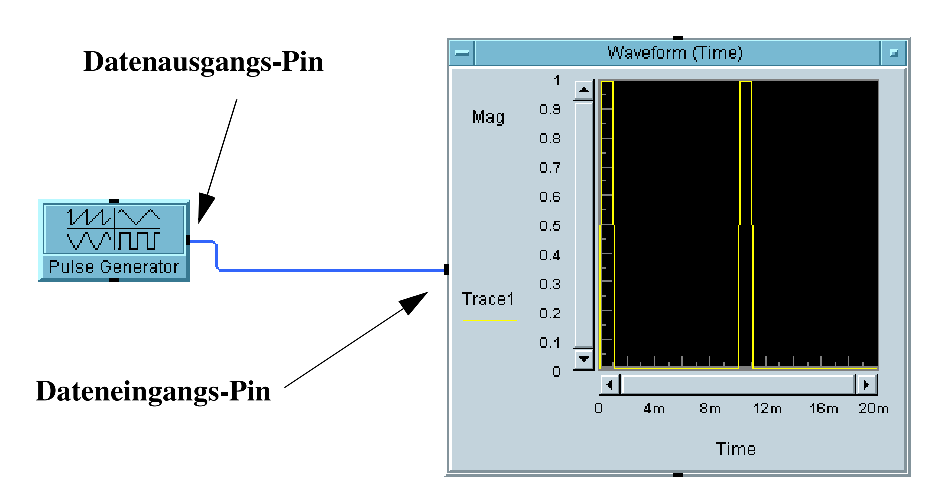


Abbildung 1-15. Erstellen von Datenlinien zwischen Objekten

Löschen von Datenlinien zwischen Objekten

1. Drücken Sie die Tastenkombination **Umschalttaste**-**Strg**, und klicken Sie die zu löschende Linie an.

   -ODER-

   Wählen Sie `Edit ⇒ Delete Line` aus, und klicken Sie die zu löschende Linie an.

Verschieben des gesamten Arbeitsbereichs

1. (Vergewissern Sie sich, dass sich mindestens ein Symbol im Arbeitsbereich befindet.) Positionieren Sie den Mauszeiger auf eine beliebige Stelle im Hintergrund des Arbeitsbereichs, drücken Sie die linke Maustaste und halten Sie sie gedrückt, und verschieben Sie den Arbeitsbereich in eine beliebige Richtung.

Hinweis Wenn Ihr Programm größer ist als der Arbeitsbereich, werden Schiebeleisten angezeigt (siehe Abbildung 1-16).

Hinweis Wenn Sie einen normalen Anschluss anklicken, kann eine Linie angezeigt werden. Verschieben Sie in diesem Fall den Zeiger auf einen offenen Bereich, und klicken Sie doppelt.

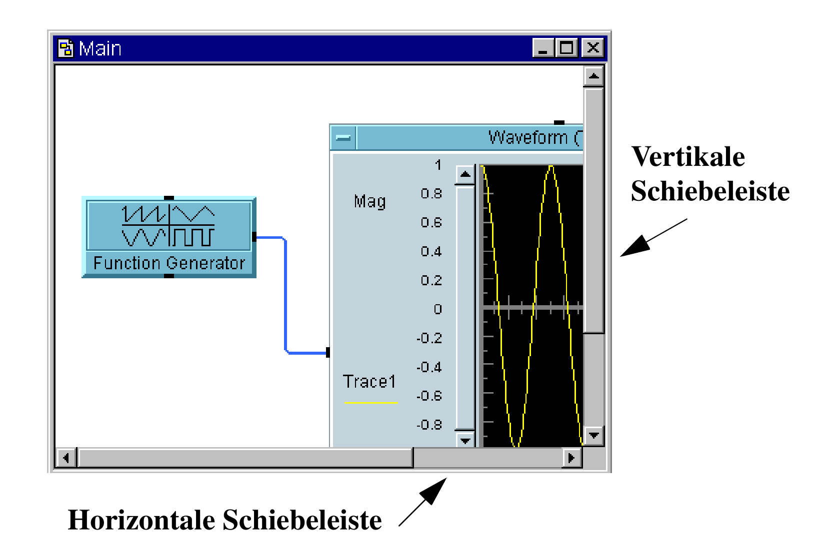


Abbildung 1-16. Schiebeleisten im Arbeitsbereich

Löschen des Arbeitsbereichs

1. Klicken Sie `Edit` ⇒ `Select All` an, und klicken Sie anschließend die Schaltfläche `Ausschneiden` in der Symbolleiste an. Dadurch werden alle Objekte im aktiven Fenster in den `Ausschneidepuffer` verschoben.

   -ODER-

   Wählen Sie `File` ⇒ `New` aus oder klicken Sie die Schaltfläche **Neu** in der Symbolleiste an. VEE fragt, ob die Änderungen gespeichert werden sollen.

   -ODER-

   Wenn Sie ein einzelnes Objekt löschen wollen, klicken Sie es an, um es zu aktivieren, und klicken Sie anschließend die Schaltfläche `Ausschneiden` in der Symbolleiste an.

Ändern der Standardeinstellungen

Über das Dialogfenster `Default Preferences` (Standardeinstellungen) können Sie die Standardeinstellungen für die VEE-Umgebung ändern.

1. Klicken Sie die Schaltfläche `Standardeinstellungen` in der Symbolleiste an.

   -ODER-

   Klicken Sie `File` ⇒ `Default Preferences` an. Das Dialogfenster `Default Preferences` wird angezeigt (siehe Abbildung 1-17).

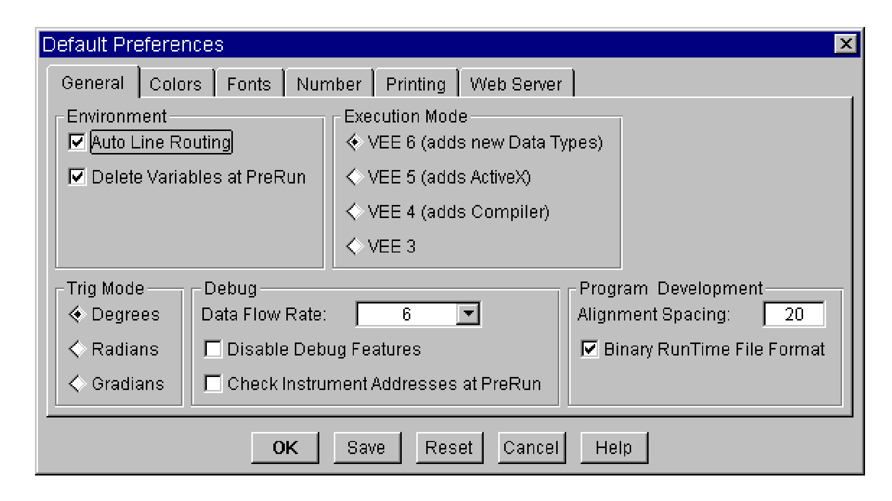


Abbildung 1-17. Default Preferences, Dialogfenster

Dieses Dialogfenster enthält *Register*, über die Sie Optionen zum Bearbeiten auswählen können.

General (Allgemein)	Das Standardregister beim Aufruf des Dialogfensters `Default Preferences` (zuvor angezeigt). Sie können die Werte der angezeigten Parameter ändern, beispielsweise `Environment` (Umgebung) und den `Execution Mode` (Ausführungsmodus).
Colors (Farben)	Ermöglicht das Anpassen der Farben in der VEE-Umgebung.
Fonts (Schriften)	Ermöglicht das Anpassen der Schriften in der VEE-Umgebung.
Number (Zahlen)	Ermöglicht das Ändern des Standardzahlenformats.
Printing (Drucken)	Zeigt die Parameterwerte für einen Drucker an.
Web Server	Ermöglicht das Aktivieren des integrierten Web-Servers zur Überwachung und Fehlerbehebung eines Programms über einen fernen Web-Browser.

Weitere Informationen können Sie über `Help` ⇒ `Contents and Index` in der VEE-Menüleiste aufrufen. Blättern Sie anschließend durch `How Do I...`, `Tell Me About...` oder `Reference`.

Informationen zu Pins und Anschlüssen

Ein VEE-Programm besteht aus den Objekten im Arbeitsbereich *und* den Linien, die diese Objekte verbinden. Die Linien, die VEE-Objekte miteinander verbinden, sind an die *Pins* der Objekte "angeschlossen". Jedes Objekt hat verschiedene Pins, wie in Abbildung 1-18 gezeigt. Abbildung 1-18 verwendet das `Formula`-Objekt als Beispiel. Sie können jedes beliebige Objekt verwenden.

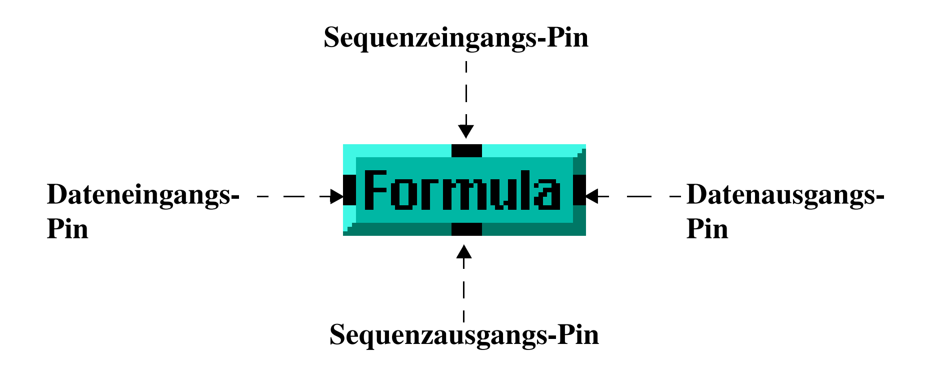


Abbildung 1-18. Daten- und Sequenz-Pins

Dateneingangs-Pin	Der Pin (bzw. die Pins) auf der linken Seite eines Objekts.
Datenausgangs-Pin	Der Pin (bzw. die Pins) auf der rechten Seite eines Objekts.
Sequenzeingangs-Pin	Der Pin an der Oberseite eines Objekts.
Sequenzausgangs-Pin	Der Pin an der Unterseite eines Objekts.

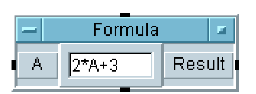
Verbinden Sie die Dateneingangs- und Ausgangs-Pins zum Transport von Daten zwischen Objekten. Standardmäßig werden die Pins von oben nach unten ausgeführt. Die Verbindungen der Sequenz-Pins sind optional. Wenn sie verbunden sind, legen sie die Reihenfolge der Ausführung fest.

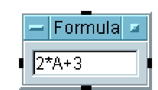
Hinweis

Weitere Informationen hierzu finden Sie unter “Verfolgen der Ereignisreihenfolge in einem Objekt” auf Seite 116.

In einer offenen Ansicht eines Objekts erscheinen die Dateineingangs- und -ausgangs-Pins als Eingangs- und Ausgangs-**Anschlüsse**. (Wenn das Objekt in der Symbolansicht angezeigt wird, klicken Sie doppelt darauf, um zur offenen Ansicht zu wechseln.) Die Anschlüsse enthalten Detailinformationen wie den Namen des Anschlusses, den Typ und den Wert der übertragenen Daten. Die Anschlussbeschriftungen sind nur in der offenen Ansicht sichtbar und nur, wenn die Option `Show Terminals` (Anschlüsse anzeigen) für dieses Objekt eingeschaltet ist (siehe `Properties`... im Objektmenü).

Abbildung 1-19 enthält beispielsweise `Formula`-Objekte. Das `Formula`-Objekt auf der linken Seite zeigt die Terminal-Beschriftungen `A` und `Result`. Für das `Formula`-Objekt auf der rechten Seite ist `Show Terminals` ausgeschaltet, und die Beschriftungen sind nicht sichtbar.







Show terminals ist ausgeschaltet

Abbildung 1-19. Show Terminals eingeschaltet am Objekt

Wählen Sie zum Ein- oder Ausschalten von `Show Terminals` die Option `Eigenschaften` im Objektmenü aus. Das Dialogfenster "Eigenschaften" zeigt ein Auswahlkästchen vor `Show Terminals` an (siehe Abbildung 1-20).

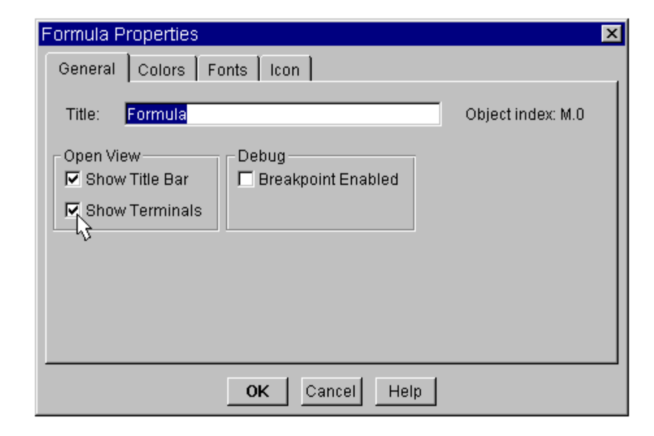


Abbildung 1-20. Verwenden des Auswahlkästchens "Show Terminals"

Klicken Sie das Auswahlkästchen an, um `Show Terminals` auszuschalten (`OFF`). Klicken Sie das Auswahlkästchen erneut an, um `Show Terminals` wieder einzuschalten (`ON`). Klicken Sie `OK` an, nachdem Sie Ihre Auswahl getroffen haben.

Einen Anschluss hinzufügen

Sie können einem Objekt Anschlüsse hinzufügen. Sie können dem `Formula`-Objekt beispielsweise einen zweiten Dateneingangsanschluss hinzufügen.

1. Öffnen Sie das Objektmenü und wählen Sie `Add Terminal` ⇒ `Data Input` aus.

   -ODER-

   Wenn `Show Terminals` eingeschaltet ist, können Sie auch mit dem Mauszeiger auf den “Anschlussbereich” (den linken Rand des Objekts in der offene Ansicht) zeigen und **Strg+A** drücken (drücken Sie **Strg** und **A** gleichzeitig).

Abbildung 1-21 zeigt das `Formula`-Objektmenü geöffnet an zum Hinzufügen eines Dateneingangsanschlusses sowie ein weiteres `Formula`-Objekt, dem bereits ein zweiter Anschluss hinzugefügt wurde. Der neue Anschluss

ist als B beschriftet. Wenn die Dateneingänge an bestimmte Funktionen gebunden sind, wie etwa bei Gerätetreibern, wird ein Menü mit diesen Funktionen angezeigt. Andernfalls haben die Anschlüsse die Namen A, B, C....

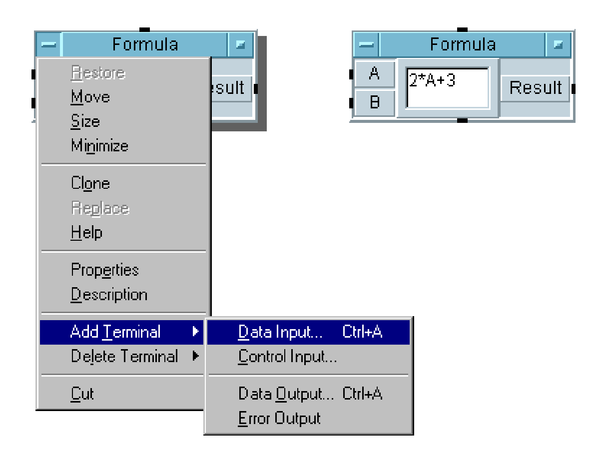


Abbildung 1-21. Einen Anschluss hinzufügen

Anschlussinformationen bearbeiten

Zum Abrufen von Informationen zu einem Anschluss klicken Sie doppelt auf den Beschriftungsbereich. Wenn Sie beispielsweise doppelt auf B klicken, wird das Dialogfenster in Abbildung 1-22 angezeigt.

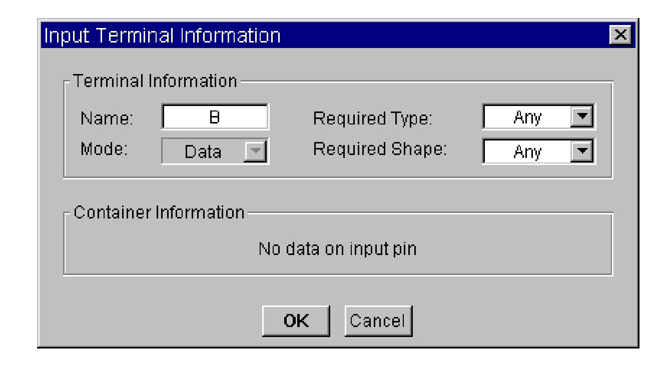


Abbildung 1-22. Anschlussinformationen abrufen

Sie können jetzt den Anschluss bearbeiten. Das Dialogfenster enthält drei Arten von Feldern:

Eingabefeld Ein Feld mit einem weißen Hintergrund, aber ohne Pfeil. Wenn Sie dieses Feld anklicken, können Sie anschließend Daten darin *eingeben*. Sie können beispielsweise `B` im Feld `Name` anklicken und den Anschluss dann umbenennen.

Statusfeld Ein Feld mit einem grauen Hintergrund; es kann nicht geändert werden. Das Feld `Mode` kann beispielsweise nicht geändert werden.

Auswahlfeld Ein Feld mit einem weißen Hintergrund, das auf der rechten Seite einen Pfeil aufweist. Durch Anklicken des Felds oder des entsprechenden Pfeils wird eine **Dropdown-Liste** angezeigt. Wenn Sie beispielsweise `Any` (oder den Pfeil) im Feld `Required Type` anklicken, können Sie einen anderen Datentyp in der Liste auswählen, indem Sie die Liste wie in Abbildung 1-23 gezeigt anklicken.

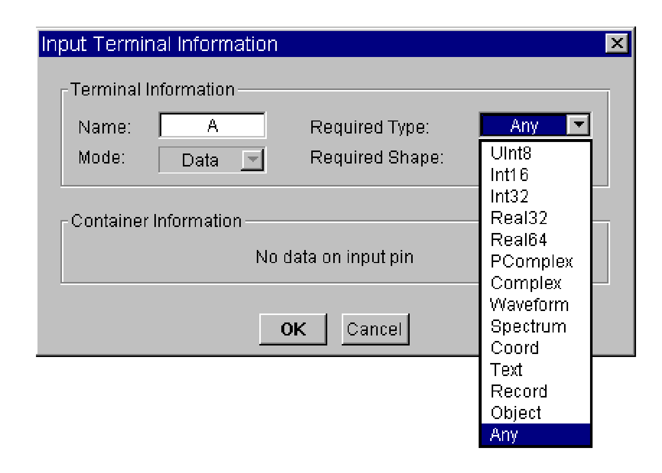


Abbildung 1-23. Verwenden des Auswahlfelds

Wenn Sie einen anderen Datentyp als `Any` für einen Dateneingangsanschluss auswählen, werden nur der angegebene Datentyp bzw. Daten, die in diesen Typ umgewandelt werden können, von dem Anschluss akzeptiert. In den meisten Fällen ist es am besten, `Required Type` und `Required Shape` auf die Angabe `Any` eingestellt zu lassen. Weitere Informationen können Sie über `Help` ⇒ `Contents and Index` in der VEE-Menüleiste aufrufen. Blättern Sie anschließend durch `How Do I`..., `Tell Me About`... oder `Reference`.

Einen Anschluss löschen

1. Öffnen Sie das Objektmenü und wählen Sie `Delete Terminal` ⇒ `Input...` oder `Delete Terminal` ⇒ `Output` aus, wählen Sie den zu löschenden Ein- oder Ausgang aus, und klicken Sie `OK` an. Abbildung 1-24 zeigt beispielsweise das Dialogfenster, das angezeigt wird, wenn Sie `Delete Terminal` ⇒ `Input...` auswählen.

   -ODER-

   Positionieren Sie den Mauszeiger auf den Anschluss und drücken Sie **Strg-D**.

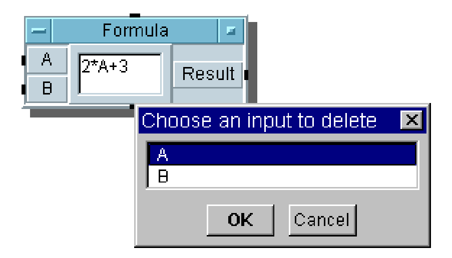


Abbildung 1-24. Das Dialogfenster "Delete Terminal"

Verbinden von Objekten zum Erstellen eines Programms

In diesem Abschnitt werden VEE Programme vorgestellt. In Übung 1-1 erstellen Sie ein VEE-Programm, drucken die VEE-Anzeige und speichern das Programm in einer Datei.

Übung 1-1: Anzeigen eines Wellenformprogramms

Ein VEE-Programm besteht aus VEE-Objekten, die in einem ausführbaren *Objektdiagramm* verbunden wurden. Das folgende Programm zeigt eine Wellenform an.

(Wenn VEE aktiv ist, löschen Sie den Inhalt des Arbeitsbereichs, indem Sie die Schaltfläche `Neu` in der Symbolleiste oder `File` ⇒ `New` verwenden. Andernfalls starten Sie VEE und fahren mit dem folgenden Schritt fort.)

1. Dokumentieren Sie das Programm. Wählen Sie `Display` ⇒ `Note Pad` aus und platzieren Sie diesen Notizblock oben in der Mitte des Arbeitsbereichs. Klicken Sie den Bearbeitungsbereich an, um einen Cursor zu erhalten, und geben Sie ein:

   ```
   Display Waveform generates a cosine waveform and sends
   it to a real time display.
   ```

   Je nach dem Bildschirm müssen Sie eventuell den `Notizblock` verkleinern. Zum Ändern der Größe öffnen Sie das Objektmenü, wählen `Size` aus, verschieben den Größenänderungs-Cursor auf eine Ecke des Objekts und ziehen. Sie können auch eine beliebige Ecke des Objekts anklicken und ziehen.)

2. Fügen Sie das Objekt `Function Generator` hinzu. Wählen Sie `Device` ⇒ `Virtual Source` ⇒ `Function Generator`, positionieren Sie den Umriss auf der linken Seite des Arbeitsbereichs, und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Ändern Sie die Frequenz auf 100, indem Sie das Feld `Frequency` anklicken und `100` eingeben.

3. Fügen Sie das Objekt `Waveform (Time)` hinzu. Wählen Sie `Display` ⇒ `Waveform (Time)` aus, und platzieren Sie das Objekt auf der rechten Seite des Arbeitsbereichs, wie in Abbildung 1-25 gezeigt.

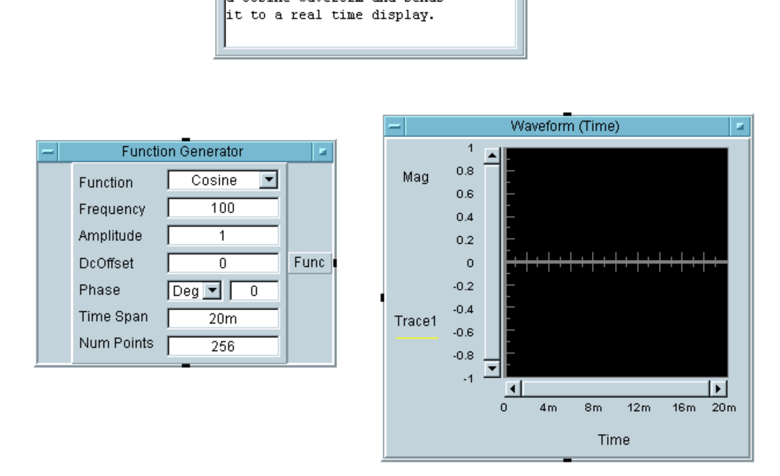


Abbildung 1-25. Erstellen eines Programms

In Abbildung 1-25 kennzeichnet die Beschriftung `Func` am Objekt `Function Generator` einen **Datenausgangs-Pin**, und die Beschriftung `Trace1` am Objekt `Waveform(Time)` kennzeichnet einen **Dateneingangs-Pin**. In VEE-Programmen verbinden Sie die Daten-Pins der Objekte und legen dadurch den Ablauf des Programms fest.

4. Vervollständigen Sie das Programm, indem Sie den Datenausgangs-Pin am `Function Generator` (neben `Func` auf der rechten Seite) mit dem Dateneingangs-Pin an der Anzeige `Waveform (Time)` (neben `Trace1` auf der linken Seite verbinden). Verschieben Sie zum Verbinden den Cursor auf einen der Pins.

   Die Form des Cursors ändert sich, wenn er in der Nähe eines Pins steht, an dem eine Verbindung zulässig ist. Klicken Sie mit der linken Maustaste, verschieben Sie den Mauszeiger auf den anderen Pin und klicken Sie erneut. Dadurch wird automatisch eine Linie zwischen den beiden Pins geführt, und das Programm ist fertig.

Versuchen Sie, eines dieser Objekte durch Ziehen seiner Titelleiste zu verschieben. (Ziehen Sie nicht an einem Pin oder einem Anschluss; dadurch wird eine Linie angezeigt.) Die Linie wird automatisch an dem logischen Pfad zwischen den beiden Objekten angelegt.

Wenn die Linien verworren erscheinen, verwenden Sie `Edit` ⇒ `Clean Up Lines` (Linien bereinigen), um die Linien in dem Programm neu zu ziehen.

Ein Programm ausführen

5. Klicken Sie zur Fortsetzung dieser Übung die Schaltfläche **Ausführen** in der Symbolleiste an, um das Programm auszuführen, oder verwenden Sie `Debug` ⇒ `Run`. Das Programm zeigt eine 100 Hz Kosinuskurve in der Anzeige `Waveform (Time)` an, wie in Abbildung 1-26 gezeigt. (Ihr Objekt hat eventuell eine andere Frequenz; dies ist jedoch für das Beispiel nicht von Bedeutung.)

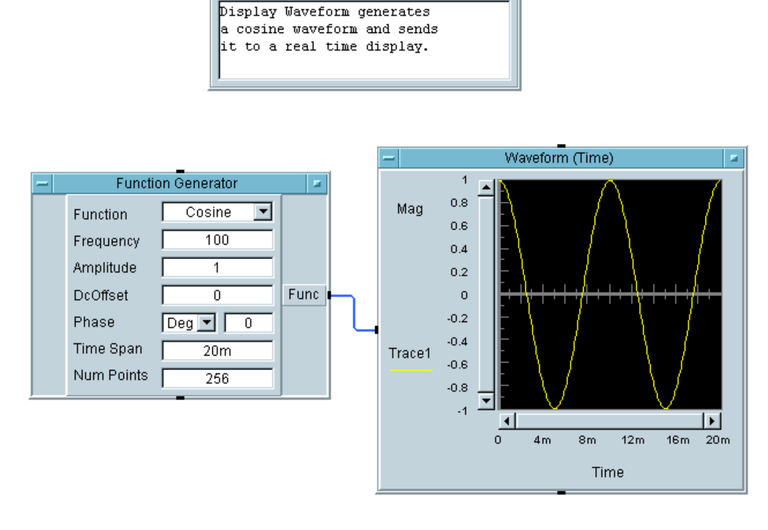


Abbildung 1-26. Ausführen eines Programms

Zusätzlich zu der Schaltfläche **Ausführen** in der Symbolleiste können Sie die Schaltflächen **Stopp**, **Pause** und **Step** in der Symbolleiste zur Steuerung des Programms verwenden. Wenn Sie ein laufendes Programm anhalten, verwenden Sie zur Fortsetzung die Schaltfläche **Fortsetzen** (an der gleichen Stelle wie **Ausführen**). Sie können mit der Schaltfläche **Schritt in** in der Symbolleiste ein Programm Objekt für Objekt ausführen.

Wenn Sie zum Ausführen des Programms aufgefordert werden, klicken Sie die Schaltfläche **Ausführen** in der Symbolleiste an oder drücken Sie **Strg**+**G**. Weitere Tastatur-Direktaufrufe sind:

Pause	**Strg**+**P**
Fortsetzen	**Strg**+**G**
Schritt in	**Strg**+**T**

Ändern von Objekteigenschaften

Sie haben gesehen, wie einige Eigenschaften eines Objekts durch Auswahl seines Objektmenüs ⇒ `Properties` geändert werden können. Sie können die wichtigsten Eigenschaften eines Objekts auch direkt in seiner offenen Ansicht ändern. Sie haben vielleicht schon bemerkt, dass das Objekt `Function Generator` zwei Arten von Feldern enthält. Ein Feld mit einem Pfeil auf der rechten Seite ist ein *Auswahlfeld*.

6. Zur Fortsetzung des Beispiels klicken Sie `Cosine` (oder den Pfeil) im Feld `Function` an. Es erscheint eine Dropdown-Liste mit Auswahloptionen. Klicken Sie `Sine` an, um die Sinusfunktion auszuwählen, wie in Abbildung 1-27 gezeigt. Beachten Sie, dass das Feld `Function` von `Cosine` in `Sine` geändert wurde.

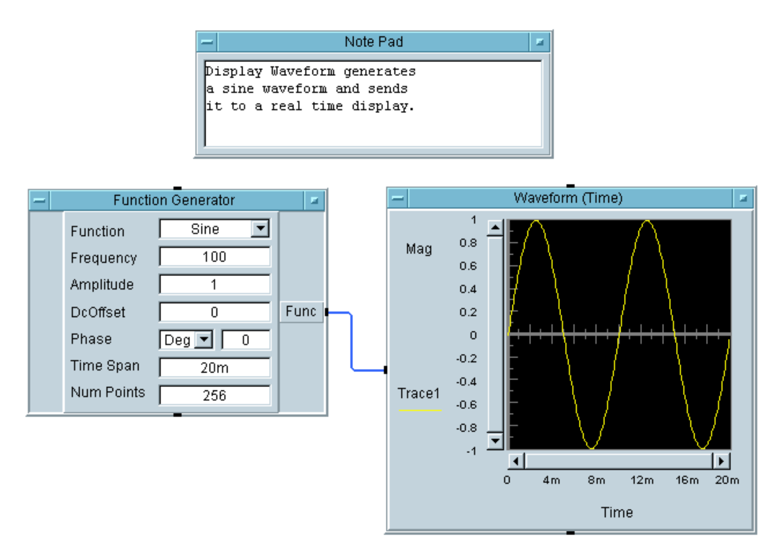


Abbildung 1-27. Ändern des Funktionsfelds in eine Sinuskurve

Einige Felder in den Dialogfeldern enthalten keine Pfeile. Dies sind die Eingabefelder, in die Sie nach dem Anklicken Daten eintippen können. Klicken Sie einfach ein Feld an, um einen Cursor anzuzeigen. Sie können die üblichen Tastatur- und Maustechniken zur Bearbeitung verwenden, um den Cursor zu verschieben und den gewünschten Wert einzugeben.

7. Klicken Sie das Feld `Frequency` rechts von dem Wert `100` an, halten Sie die Maustaste gedrückt und verschieben Sie die Maus nach links, um die letzte `0` hervorzuheben, wie in Abbildung 1-28 gezeigt.

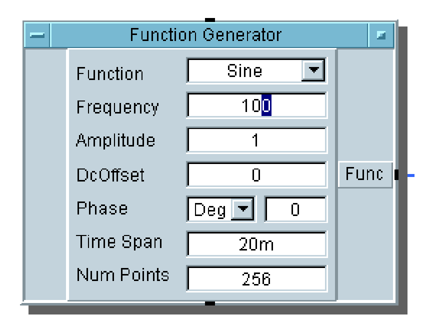


Abbildung 1-28. Hervorheben einer Zahl im Frequenzfeld

8. Drücken Sie die Taste **Entf**, um die letzte `0` zu löschen. Der Wert für `Frequency` beträgt jetzt `10`. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 1-29.

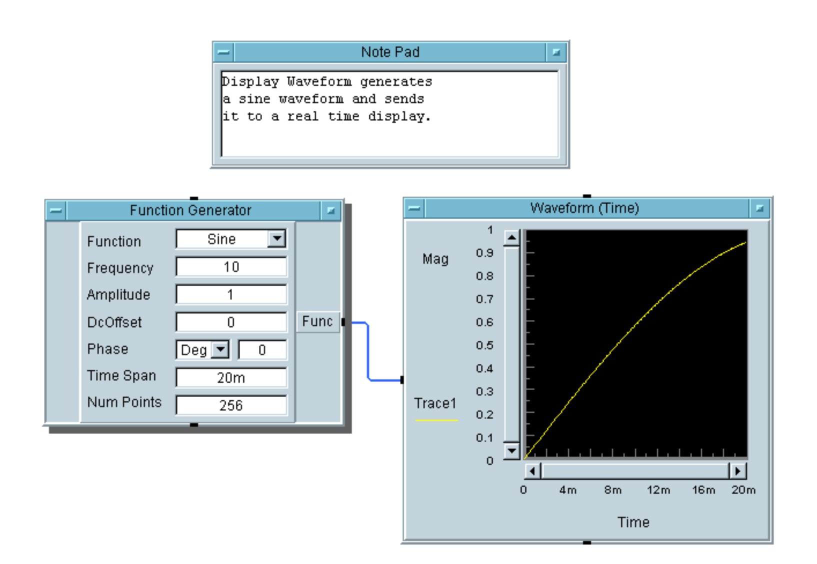


Abbildung 1-29. Beispiel: Ändern des Frequenzfelds in 10 Hz

Die angezeigte Wellenform ist jetzt eine 10 Hz-Sinuskurve. Sie können einige Objektparameter wie folgt ändern:

- Klicken Sie `Deg` (oder den Pfeil) im Objekt `Function Generator` an, und ändern Sie die Phaseneinheit in `Rad`. Klicken Sie als nächstes das Feld `Phase` an und geben Sie den Wert `PI` ein. Führen Sie das Programm erneut aus und notieren Sie die Phasenverschiebung in der angezeigten Wellenform. Ändern Sie anschließend den Wert `Phase` zurück in `0` und die Einheit zurück in `Deg`.

- Die Y-Achsenbegrenzungen des Objekts `Waveform (Time)` sind voreingestellt auf die Werte `-1` bis `1`. Klicken Sie Namen der Y-Achse `Mag` an, um ein Dialogfenster zu öffnen, mit dem Sie die Einstellungen ändern können. Klicken Sie die Felder für das `Maximum` und `Minimum` an, um die Begrenzungen in `2` und `-2` zu ändern. Die Wellenform wird mit den neuen Begrenzungen angezeigt. Klicken Sie zum Ändern vergleichbarer Parameter für die Skala der X-Achse `Time` an.

Ausdrucken der Anzeige

9. Wählen Sie zur Fortsetzung des Beispiels und zum Drucken der Anzeige `File` ⇒ `Print Screen` aus. Unter Windows wird das in Abbildung 1-30 dargestellte Dialogfenster angezeigt.

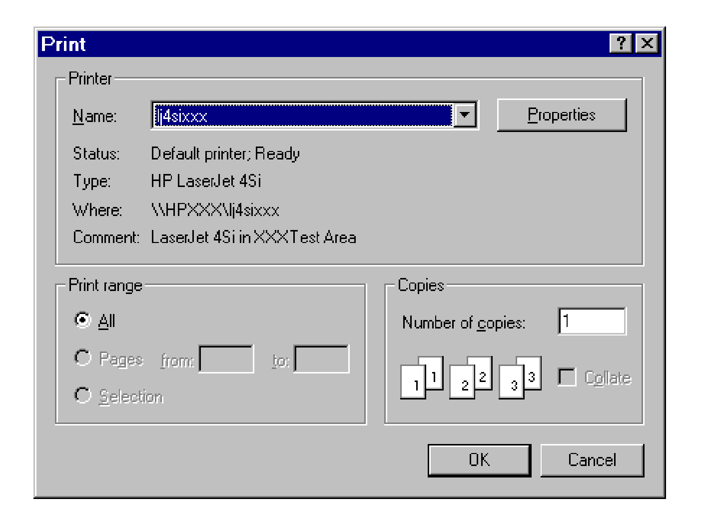


Abbildung 1-30. Drucken der Anzeige

Wenn Sie `OK` anklicken, druckt VEE die Anzeige auf dem in dem Dialogfenster angegebenen Standarddrucker aus. Sie können einen anderen Drucker auswählen, den Druckbereich ändern und die Anzahl der zu druckenden Exemplare eingeben. Klicken Sie die Schaltfläche `Properties` für weitere Auswahloptionen an. Für verschiedene Druckertreiber können unterschiedliche Dialogfenster angezeigt werden. Weitere Informationen zur Verwendung von Windows-Dialogfenstern finden Sie in der *Microsoft Windows Hilfe*.

Unter HP-UX wird das in Abbildung 1-31 dargestellte Dialogfenster angezeigt.

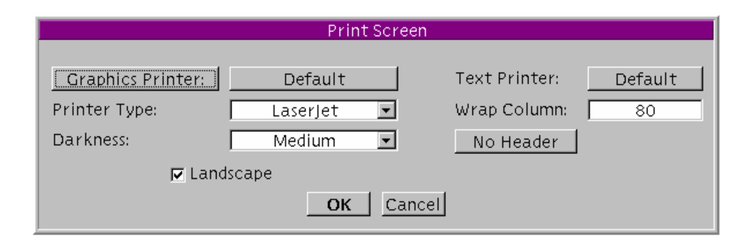


Abbildung 1-31. Dialogfenster "Anzeige drucken"

Wenn Sie `OK` anklicken, druckt VEE die Anzeige auf dem ausgewählten Drucker. Sie können über dieses Dialogfenster einen Grafik- oder Textdrucker auswählen. Außerdem können Sie vor dem Drucken die Konfiguration dieser Geräte ändern.

Direktaufruf: Klicken Sie die Schaltfläche `Bildschirm drucken` in der Symbolleiste an, um die Anzeige direkt zu drucken.

Speichern eines Programms

Sie können ein Programm jederzeit speichern. (Der Inhalt des Arbeitsbereichs kann gespeichert werden, ob es sich dabei um ein Programm handelt oder nicht.)

10. Wählen Sie zur Fortsetzung des Beispiels `File ⇒ Save As...` aus, und füllen Sie die Felder in dem Dialogfenster aus.

    Daraufhin wird das Dialogfenster `Save File` angezeigt. Abbildung 1-32 zeigt das PC-Format dieses Dialogfensters.

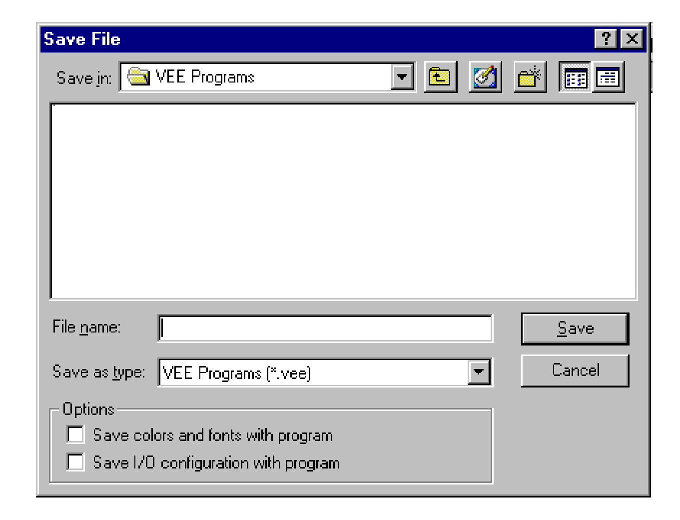


Abbildung 1-32. Das Dialogfenster "Save File" (PC)

11. Standardmäßig speichert VEE für Windows Dateien im Unterverzeichnis `VEE Programs` des Verzeichnisses `My Document`. Geben Sie zum Speichern des aktuellen Programms den Namen `simple-program` im Feld `File name` ein und klicken Sie `Save` an. Wenn Sie die Erweiterung `.vee` nicht angeben, wird sie von VEE automatisch hinzugefügt.

Hinweis

In VEE für Windows können Sie die bei Windows 95, Windows 98, Windows 2000 und Windows NT 4.0 zulässigen langen Dateinamen verwenden.

Im PC-Dialogfenster `Save File` können Sie wie folgt Änderungen an den verschiedenen Fenstern vornehmen:

Save in (Speichern in)	Sie können das Verzeichnis oder das Laufwerk wechseln, indem Sie das entsprechende Drop-down-Menü öffnen. Klicken Sie doppelt auf einen Ordner, um ihn zu öffnen.
File name (Dateiname)	Geben Sie einen Dateinamen Ihrer Wahl ein.

Save as type (Speichern als Typ)	VEE-Programme werden normalerweise mit der Erweiterung `.vee` gespeichert; Sie können den Dateityp jedoch auch ändern. Wenn Sie einen Dateinamen ohne Erweiterung eingeben, wird die Erweiterung `.vee` automatisch hinzugefügt.
Save colors and fonts with program (Farben und Schriften mit dem Programm speichern)	(Optional) Wenn Sie die Farben und Schriften in dem Programm über das Menü `Default Preferences` geändert haben und diese Farben und Schriften auch dann statt der Standardangaben verwendet werden sollen, wenn andere Benutzer das Programm laden, klicken Sie diese Option an. Ist diese Option markiert, speichert VEE die vorgenommenen Änderungen an der Standardkonfiguration als Teil des Programms.
Save I/O configuration with program (E/A-Konfiguration mit Programm speichern)	(Optional) Wenn Sie ein Gerät im `Instrument Manager` konfiguriert haben und Ihre Konfiguration des Geräts auch anderen Benutzern des Programms beim Laden zur Verfügung stehen soll, sollten Sie diese Option verwenden. Wenn diese Option markiert ist, speichert VEE die E/A-Konfiguration als Teil des Programms.

Hinweis

Wenn Sie mit der Evaluation Kit-Software arbeiten, erlaubt VEE das Speichern von Programmen nur in einer einzigen Datei `EVAL.VEE`; Sie können für die verschiedenen Beispiele diese Datei überschreiben.

Informationen zur HP-UX-Version des Dialogfensters `Save File` finden Sie in Abbildung 1-33.

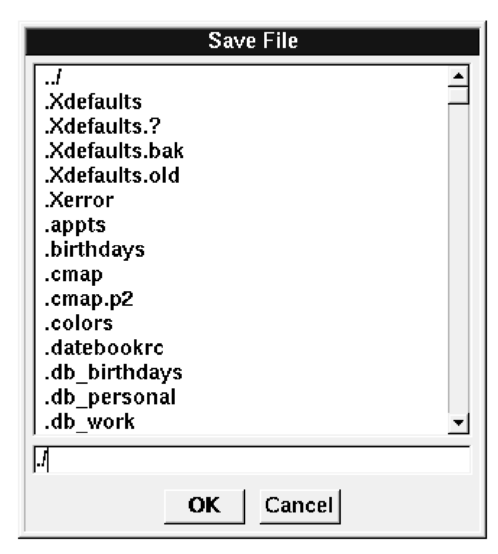


Abbildung 1-33. Das Dialogfenster "Save File" (UNIX)

Standardmäßig speichert VEE für HP-UX Dateien in dem Verzeichnis, von dem aus Sie VEE aufgerufen haben. Geben Sie zum Speichern des aktuellen Programms einen Namen ein (z. B. `simple-program.vee`) und klicken Sie `OK` an.

Wenn Sie die Datei in einem anderen Verzeichnis speichern wollen, löschen Sie die Zeichen `./` mit der **Rücktaste**, geben Sie den Dateinamen mit dem vollständigen Pfad ein, und klicken Sie `OK` an. In HP-UX müssen Sie die Erweiterung `.vee` mit dem Namen angeben.

Tip: Sie können einen in einem Dialogfenster eingetragenen Wert schnell und einfach ersetzen, indem Sie mit der Maus klicken und den Mauszeiger über den Eintrag ziehen, um ihn hervorzuheben. Sie können den Eintrag auch hervorheben, indem Sie doppelt in das Eingabefeld klicken. Anschließend können Sie den neuen Eintrag eintippen und `OK` anklicken.

Hinweis

Zum erneuten Speichern des Programms mit dem gleichen Namen klicken Sie die Schaltfläche `Speichern` an, oder drücken Sie zu einem beliebigen Zeitpunkt **Strg+S** (`File` ⇒ `Save`). Es empfiehlt sich, bei der Entwicklung eines Programms die Dateien häufig zu speichern. Wenn Sie ein geändertes Programm unter einem anderen Namen speichern wollen, drücken Sie **Strg+W**, oder wählen Sie `File` ⇒ `Save As` aus.

Beenden von Agilent VEE

12. `Wählen Sie File ⇒ Exit` aus, um das VEE-Anwendungsfenster zu schließen.

Direktaufruf: Drücken Sie **Strg-E** zum Beenden von VEE, oder klicken Sie die Schaltfläche x am rechten Rand der Titelleiste an.

Sie müssen eventuell die folgenden Techniken verwenden, wenn VEE nicht mehr auf die Eingabe über die Maus oder die Tastatur reagiert:

In Windows 95 und Windows 98	Drücken Sie **Strg-Alt-Entf**. Daraufhin wird ein Fenster mit verschiedenen Optionen angezeigt. Gehen Sie in MS Windows anhand der Anleitungen in diesem Fenster vor, oder klicken Sie `Task beenden` an.
In Windows NT 4.0 und Windows 2000	Drücken Sie **Strg-Alt-Entf** und klicken Sie die Schaltfläche `Task Manager` an. Wählen Sie `VEE` in der Liste der Anwendungen aus und klicken Sie `Task beenden` an.
In HP-UX	Sie müssen den Prozess mit “kill” beenden (dies entspricht der UNIX-Terminologie).

1. Geben Sie `ps -ef | grep vee` bei einer HP-UX-Eingabeaufforderung ein, um die Prozess-Identifikationsnummer (PIN) zu ermitteln. Am Ende der Anzeige finden Sie eine Zeile mit dem Eintrag `veetest`. Die Nummer im Anschluss an Ihre Anmeldekennung ist die PIN-Nummer, die Sie eingeben müssen. Die Anzeige könnte beispielsweise die Angabe `johnj` *nummer*`... veetest` enthalten.

2. Geben Sie `kill -9` *nummer* ein, um die VEE-Anwendung zu stoppen. Anschließend können Sie `veetest` eingeben, um von vorn zu beginnen.

Erneut Starten von Agilent VEE und Ausführen eines Programms

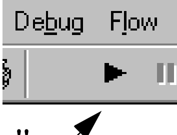
1. PC: Klicken Sie in Windows `Start` ⇒ `Programs` ⇒ `Agilent VEE Pro 6.0` an.

   HP-UX: Geben Sie von Ihrem Heimverzeichnis aus ein: `veetest`. (Die ausführbare Datei wird bei der Installation mit `/usr/bin/veetest` verknüpft; diese Angabe sollte in der PATH-Variablen enthalten sein. Ist dieses Verzeichnis nicht in der PATH-Variablen angegeben, müssen Sie in dieses Verzeichnis wechseln. Wenn VEE in einem anderen Verzeichnis installiert wurde, müssen Sie den gesamten Pfad eingeben.)

2. `Wählen Sie File` ⇒ `Open` aus, und füllen Sie das Dialogfenster `Open File` aus.

   Das Format ist das gleiche wie für das Dialogfenster `Save File`. In VEE für Windows lautet das Standardverzeichnis für Benutzerprogramme `VEE_USER`, sofern Sie bei der Installation kein anderes Verzeichnis angegeben haben. VEE öffnet das Programm im Hauptfenster.

3. Klicken Sie die Schaltfläche **Ausführen** an. Diese Schaltfläche sieht aus wie eine kleine Pfeilspitze; sie befindet sich in der Symbolleiste unterhalb des Menüs `Debug`, wie in Abbildung 1-34 gezeigt.



Die Schaltfläche "Ausführen"

Abbildung 1-34. Die Schaltfäche "Ausführen" in der Symbolleiste

Hinweis

PC: Der Befehl `vee.exe -r` *Dateiname* startet VEE in Windows und führt automatisch das mit *Dateiname* angegebene Programm aus. Sie können beispielsweise ein Symbol im Windows-Desktop erstellen und `Eigenschaften` ⇒ `Tastenkombination` für dieses Symbol auf die Ausführung eines

bestimmten VEE-Programms einstellen. Ein Bediener kann dann doppelt auf das Symbol im Desktop klicken, um VEE zu starten und automatisch ein bestimmtes Programm auszuführen. Weitere Informationen finden Sie in der Windows-Hilfe zu Pfaden für Befehle und Eingabeaufforderungen.

HP-UX: Der Befehl `veetest -r` *filename* startet VEE und führt automatisch das mit *Dateiname* angegebene Programm aus. (Wenn das VEE-Verzeichnis nicht in dem Pfad enthalten ist, müssen Sie den vollständigen Pfad eingeben, normalerweise `/usr/bin/veetest -r` *Dateiname*.)

Verwalten mehrerer Fenster im Arbeitsbereich

Bis hierher beziehen sich die meisten Erläuterungen auf den Arbeitsbereich im Hauptfenster. Große VEE-Programme können jedoch mehrere Fenster innerhalb des Hauptfensters enthalten. Ein Programm kann beispielsweise von Ihnen definierte Objekte enthalten, etwa `UserObjects` und `UserFunctions`. (Sie können sich `UserObjects` und `UserFunctions` wie Unterprogramme des Hauptprogramms vorstellen. `UserObjects` und `UserFunctions` werden im Abschnitt “Erstellen eines UserObject” auf Seite 82 in Kapitel 2, “Agilent VEE Programmiertechniken” ausführlich beschrieben). Sie werden hier nur erwähnt, um zu zeigen, wie VEE die Verwaltung von Programmen mit mehreren Fenstern ermöglicht.

Abbildung 1-35 zeigt ein Programm mit vier Fenstern. Jedes Fenster enthält ein Symbol (mit Menübefehlen), einen Titel und drei Schaltflächen: Symbolgröße, Vollbildgröße und Schließen. Die Anzeige eines Fensters in Vollbildgröße bewirkt, dass das Fenster den gesamten verfügbaren Platz im VEE-Arbeitsbereich belegt. Die Anzeige in Symbolgröße zeigt das Fenster als Symbol am unteren Rand des VEE-Arbeitsbereichs an. Durch das Schließen eines Fensters wird es aus dem Arbeitsbereich entfernt. VEE hebt die Titelleiste des jeweils aktiven Fensters hervor.

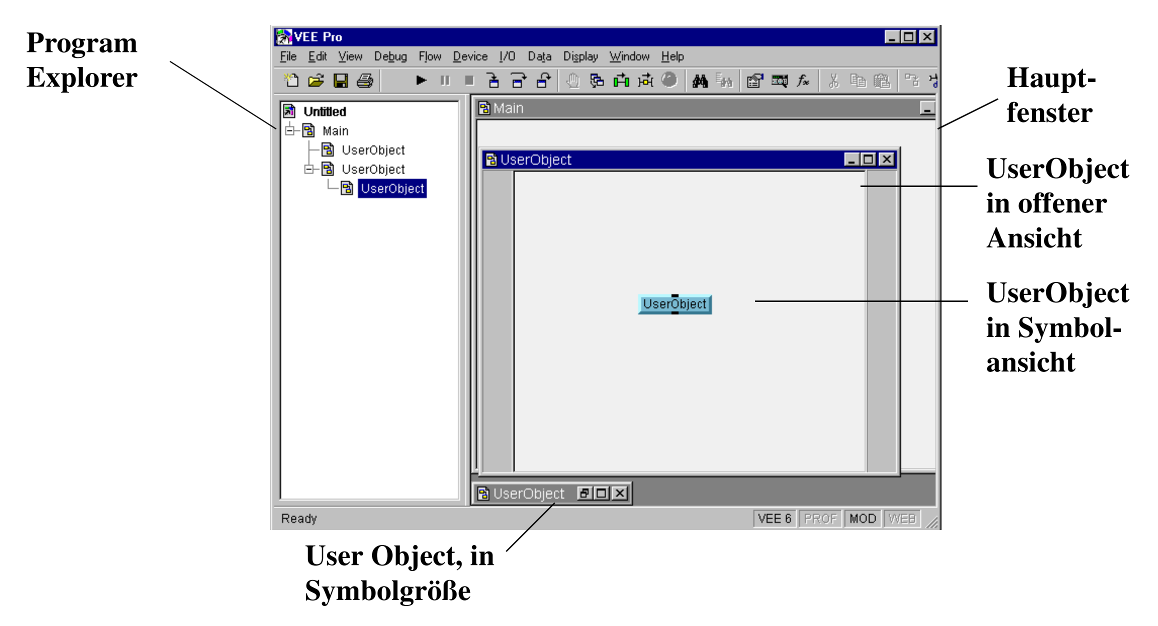


Abbildung 1-35. Mehrere Fenster im Arbeitsbereich

Abbildung 1-35 zeigt, wie der Program Explorer die Hierarchie des Programms auflistet. Diese integrierte modulare Struktur ermöglicht einen einfachen Zugriff auf alle Teile des Programms.

Wenn der Program Explorer nicht angezeigt wird, klicken Sie `View` ⇒ `Program Explorer` an. Standardmäßig wird der Program Explorer angezeigt. Wenn Sie das Häkchen entfernen und die Schaltfläche `Save` unter `File` ⇒ `Default Preferences` anklicken, erscheint der Program Explorer beim nächsten Starten von VEE nicht mehr.

Sie können das Hauptfenster jederzeit in den Vordergrund bringen, indem Sie es anklicken oder auf sein Symbol im Program Explorer doppelklicken.

Hinweis

Wenn Sie das Hauptfenster in VEE schließen, können Sie es erneut anzeigen, indem Sie `View` ⇒ `Main` auswählen.

Die Arbeitsweise von Agilent VEE-Programmen

In VEE wird der allgemeine Ausführungsablauf in einem Programm als **Weitergabe** (propagation) bezeichnet. Die Weitergabe in einem Programm ist *nicht* durch die geografische Position der Objekte im Programm bestimmt, sondern durch die Verbindungen zwischen den Objekten. Die Weitergabe wird in erster Linie bestimmt durch den **Datenfluss**, der wiederum davon abhängt, wie die Datenein- und -ausgangs-Pins der Objekte miteinander verbunden sind.

Hinweis

In anderen Programmiersprachen wie C, BASIC oder Pascal wird die Reihenfolge, in der Programmanweisungen ausgeführt werden, von einer Reihe von Sequenz- und Auswahlregeln bestimmt. Im Allgemeinen werden die Anweisungen in der Reihenfolge ausgeführt, in der sie im Programm aufgeführt sind, es sei denn, bestimmte Anweisungen bewirken eine Verzweigung zu einer anderen Anweisung oder in einen Code-Thread.

Die Regeln des Datenflusses in einem VEE-Programm lauten wie folgt:

- *Daten fließen von links nach rechts durch ein Objekt* . Dies bedeutet, dass bei allen Objekten mit Daten-Pins die linken Daten-Pins dem Eingang entsprechen und die rechten Daten-Pins dem Ausgang.
- *Alle Dateneingangs-Pins in einem Objekt müssen angeschlossen werden*; Andernfalls tritt bei der Ausführung des Programms ein Fehler auf.
- *Ein Objekt wird erst ausgeführt, wenn alle seine Dateneingangs-Pins neue Daten empfangen haben*.
- *Die Ausführung eines Objekts wird erst beendet, wenn alle angeschlossenen und relevanten Datenausgangs-Pins aktiviert wurden*.

In VEE können Sie die Reihenfolge der Ausführung mit den Sequenzein- und -ausgangs-Pins ändern. Sie brauchen die Sequenz-Pins normalerweise nicht zu verwenden; diese sind nur in Sonderfällen erforderlich. *Im Allgemeinen ist es das Beste, auf die Verwendung der Sequenz-Pins ganz zu*

verzichten. Wenn möglich, lassen Sie den Datenfluss die Ausführung des Programms steuern.

Übung 1-2: Anzeige von Datenfluss und Weitergabe

Öffnen Sie das zuvor erstellte Programm, um zu sehen, wie der Datenfluss funktioniert. Öffnen Sie das Programm `simple-program.vee` durch Anklicken der Schaltfläche `Öffnen` in der Symbolleiste. (Das Programm `simple-program.vee` ist im Abschnitt “Anzeigen eines Wellenformprogramms” auf Seite 55 beschrieben.) Führen Sie jetzt das Programm aus. Es sollte wie in Abbildung 1-36 dargestellt werden; allerdings können für die Parameter andere Werte eingetragen sein.

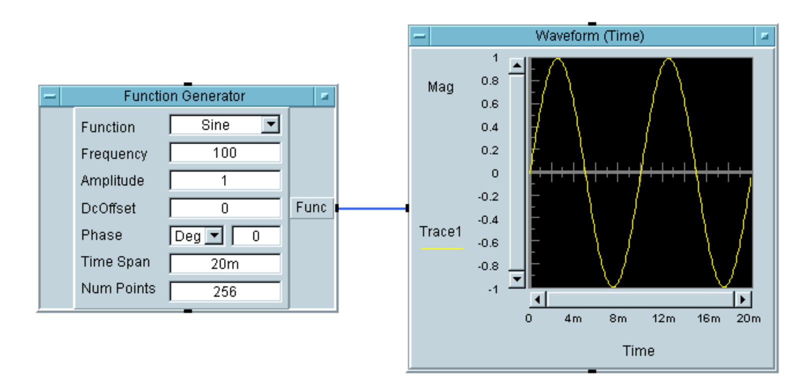


Abbildung 1-36. Typische Anzeige von simple-program.vee

Der Datenausgangs-Pin des Objekts `Function Generator` ist mit dem Dateneingangs-Pin des Objekts `Waveform (Time)` verbunden. Bei der Ausführung des Programms wird das Objekt `Waveform (Time)` nicht ausgeführt, bevor es Daten aus dem Objekt `Function Generator` empfängt. Dies ist ein einfaches Beispiel des Datenflusses.

Übung 1-3: Hinzufügen eines Noise Generator

Fügen Sie eine “gestörte Sinuskurve” hinzu, indem Sie ein Objekt `Noise Generator` dem Programm `simple-program.vee` hinzufügen, wie in Abbildung 1-37 gezeigt.

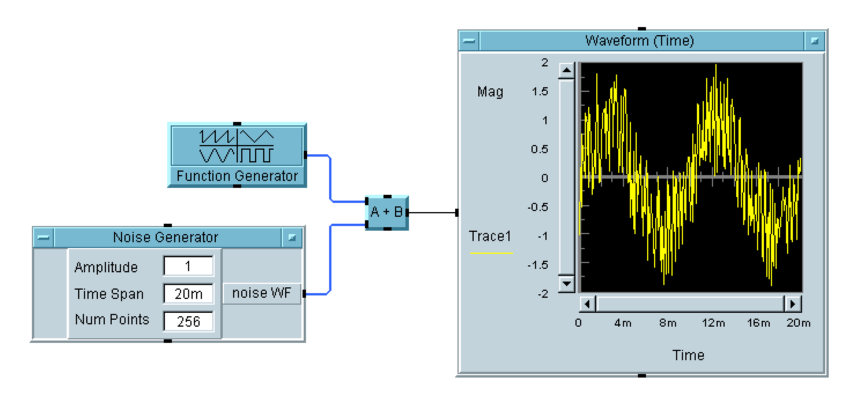


Abbildung 1-37. Beispiel: Hinzufügen eines Noise Generator-Objekts

Hinweis

Die VEE -Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter `Help` ⇒ `Open Example...` ⇒ `Manual` ⇒ `UsersGuide` enthalten.

1. Löschen Sie die Linie, die die Objekte `Function Generator` und `Waveform (Time)` in dem ursprünglichen Programm verbindet. Klicken Sie die Schaltfläche **Linie löschen** in der Symbolleiste und anschließend die Linie an. Oder drücken Sie die Tastenkombination **Umschalttaste+Strg**, halten Sie diese Tasten gedrückt, und klicken Sie die Linie an.

2. Zeigen Sie den `Function Generator` in Symbolgröße an.

3. Fügen Sie das Objekt `Noise Generator` hinzu (`Device` ⇒ `Virtual Source` ⇒ `Noise Generator`).

4. Fügen Sie das Objekt `A+B` mit `Device` ⇒ `Function & Object Browser` hinzu.

   Der `Function & Object Browser` ist in Abbildung 1-38 dargestellt. Wählen Sie für `Type` die Option `Operators` aus. Für `Category` wählen Sie `Arithmetic` aus. Für `Operators` wählen Sie + aus.) Klicken Sie `Create Formula` an und platzieren Sie das Objekt im Arbeitsbereich

zwischen den Objekten `Function Generator` und `Waveform Time)`. Zeigen Sie das Objekt `A+B` in Symbolgröße an.

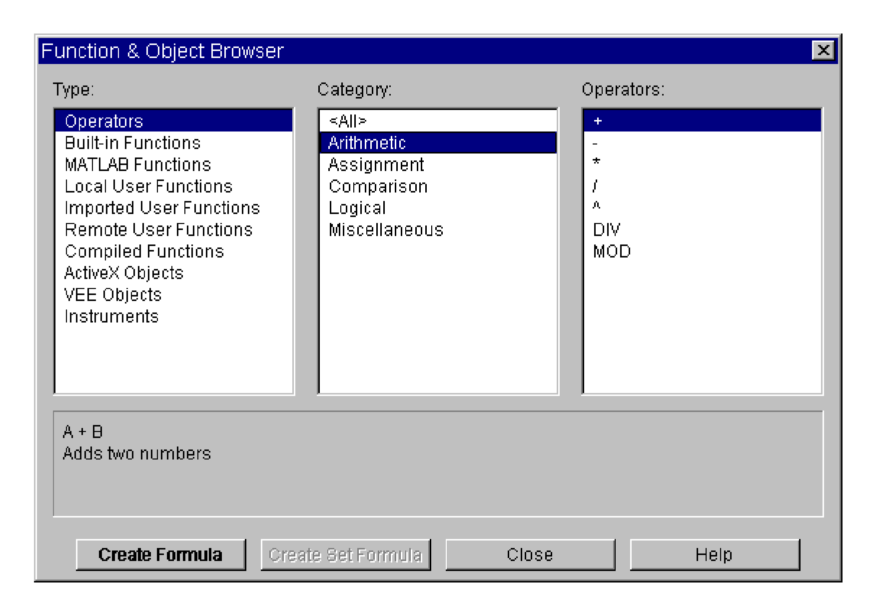


Abbildung 1-38. Function and Object Browser

5. Verbinden Sie die Ein- und Ausgangs-Pins wie in Abbildung 1-37 dargestellt.

6. Führen Sie das Programm aus.

   Beachten Sie, dass das Objekt `A+B` nicht ausgeführt wird, bevor die Objekte `Function Generator` und `Noise Generator` ausgeführt werden. Es spielt jedoch keine Rolle, ob der `Function Generator` oder der `Noise Generator` zuerst ausgeführt wird; das Ergebnis ist das Gleiche.

   Sobald beide Eingangs-Daten-Pins von `A+B` Daten empfangen, wird das Objekt `A+B` ausgeführt; dabei werden die beiden Signale summiert und das Ergebnis an das Objekt `Waveform (Time)` ausgegeben.

Hinweis Der Datenfluss in einem VEE-Programm bestimmt seine Ausführung.

Schalten Sie zur Anzeige der Ausführungsreihenfolge die `Debug`-Befehle `Show Execution Flow` (Ausführungsfluss anzeigen) und `Show Data Flow` (Datenfluss anzeigen) ein, oder klicken Sie die entsprechenden Schaltflächen in der Symbolleiste an. Führen Sie das Programm erneut aus. Jedes Objekt wird bei seiner Ausführung hervorgehoben, und eine kleine quadratische Markierung wird entlang der Linien verschoben, um den Datenfluss anzuzeigen.

Hinweis

`Show Execution Flow` und `Show Data Flow` können gleichzeitig oder einzeln aktiviert werden; klicken Sie hierzu die entsprechenden Schaltflächen in der Symbolleiste oder die entsprechenden Befehle im `Debug`-Menü an. Normalerweise sollten Sie diese Befehle ausschalten, da sie die Ausführung des Programms deutlich verlangsamen.

Übung 1-4: Hinzufügen eines Amplituden-Eingangs und eines Real64 Slider

Fügen Sie einen Amplituden-Eingang und einen Real64 Slider dem `simple-program.vee` hinzu.

1. Klicken Sie das Objektmenü an oder drücken Sie **Strg+A**, während sich der Mauszeiger im “Anschlussbereich” auf der linken Seite des `Noise Generator` befindet. Das Dialogfenster erscheint, in dem Sie einen Eingang hinzufügen können (siehe Abbildung 1-39).

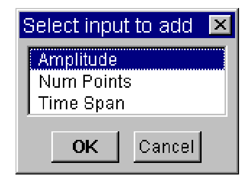


Abbildung 1-39. Beispiel: Hinzufügen von Eingangsanschlüssen

2. Wählen Sie `Amplitude`, indem Sie `OK` anklicken — es erscheint ein Eingangsanschluss `Amplitude`.

Da das Objekt `Noise Generator` jetzt einen Amplituden-Eingangs-Pin enthält, können Sie diese Daten als reale Zahl eingeben. VEE bietet ein Objekt, das diese Aufgabe erleichtert: den `Real64 Slider` (Real64-Schieberegler) im Menü `Daten`. (Sie können auch das Objekt `Real64 Constant` oder einen `Real64 Knob` verwenden.)

3. Fügen Sie ein Objekt `Real64 Slider` hinzu (`Data` ⇒ `Continuous` ⇒ `Real64 Slider`) und verbinden Sie seinen Datenausgangs-Pin mit dem Anschluss `Amplitude`, wie in Abbildung 1-40 gezeigt. Führen Sie das Programm aus.

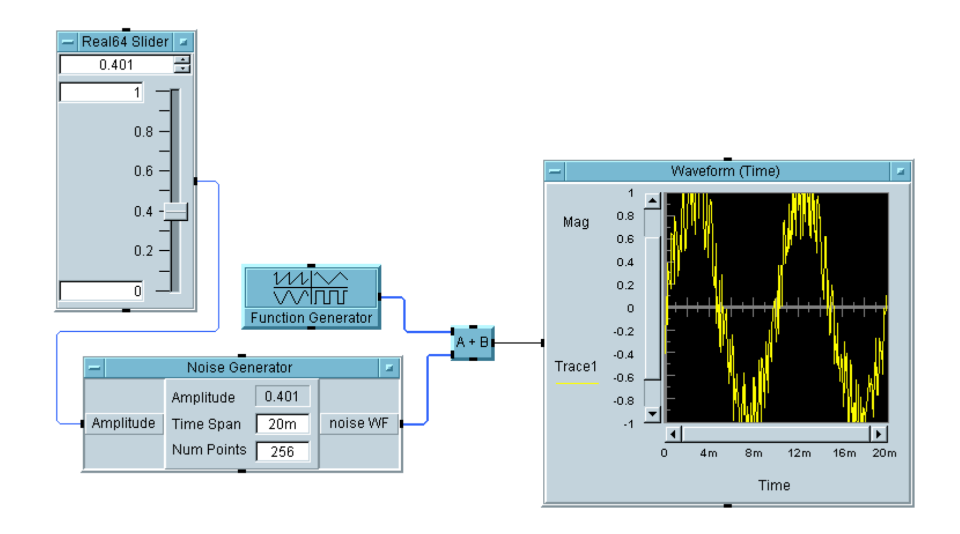


Abbildung 1-40. Beispiel: Hinzufügen eines Real64 Slider-Objekts

Versuchen Sie, die Amplitude der Störung zu ändern, indem Sie den Schieberegler auf das Objekt `Real64 Slider` ziehen. Die Amplitude der Störung ändert sich nicht, bis Sie das Programm ausführen. Der Störfaktor der angezeigten Wellenform hängt vom Ausgabewert des `Real64 Slider` ab.

Auch hier bestimmt der Datenfluss die Reihenfolge der Ausführung. Der `Noise Generator` kann nicht ausgeführt werden, bis der `Real64 Slider` ausgeführt wird. Das Objekt `A+B` kann nicht ausgeführt werden, bis sowohl der `Function Generator` als auch der `Noise Generator` ausgeführt werden; es spielt jedoch keine Rolle, welches dieser Objekte zuerst ausgeführt wird. Das Objekt `Waveform (Time)` wird erst ausgeführt, nachdem das Objekt `A+B` ausgeführt wurde.

Hinweis

Sie können den Wert eines Ausgangs anzeigen, indem Sie mit der Maus auf die Linie zeigen. Wenn Sie beispielsweise auf die Linie vom Objekt `Real64 Slider` zum Objekt `Noise Generator` zeigen, wird der Wert `0.401` angezeigt. Beachten Sie, dass der Wert der Linie (`0.401`) dem im `Real64 Slider` angezeigten Wert entspricht, wie in Abbildung 1-41 gezeigt. (Beachten Sie, dass die Objekte in der Symbolansicht angezeigt werden.)

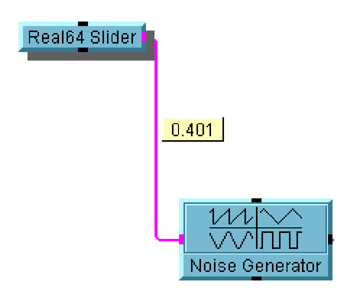


Abbildung 1-41. Anzeigen des Werts an einen Ausgangs-Pin

4. Speichern Sie das Programm erneut in `simple-program.vee`. Im nächsten Kapitel werden Sie diesem Programm einige weitere Funktionen hinzufügen.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie zum nächsten Kapitel übergehen.

- Anzeigen der Online-Hilfedokumentation von der Haupt-Menüleiste und den Objektmenüs aus.
- Starten von VEE.
- Erkennen der Haupt-Menüleiste, der Schaltflächen der Symbolleiste, des Arbeitsbereichs und der Statusleiste.
- Erläutern des Program Explorer und seiner Funktion.
- Auswählen der Menüelemente im Hauptmenü und in den Objektmenüs.
- Ausführen der folgenden Operationen mit einem Objekt: Verschieben, Umbenennen, Anzeigen als Symbol, Erweitern, Ändern der Größe, Auswählen, Aufheben der Auswahl, Löschen, Klonen etc.
- Verschieben des Arbeitsbereichs, Löschen des Arbeitsbereichs und Verwalten mehrerer Fenster.
- Erkennen der Daten- und Sequenz-Pins an einem Objekt und Erläuterung ihrer Aufgabe.
- Untersuchen der Terminals und Ändern der Namen.
- Verbinden der Objekte zum Erstellen eines Programms für die Simulation von Wellenform-Daten.
- Erstellen, Ausführen, Drucken und Speichern eines Programms.
- Beenden von VEE und erneut Öffnen eines Programms.
- Erläutern, wie Daten durch ein VEE-Programm fließen.

2

Agilent VEE Programmiertechniken

Agilent VEE Programmiertechniken

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Erstellen eines `UserObject`
- Hinzufügen eines Dialogfensters für die Benutzereingabe
- Verwenden der Datendateien
- Erstellen von Fensteransichten (Benutzerschnittstelle)
- Mathematisches Verarbeiten von Daten
- Kommunikation mit Instrumenten
- Dokumentieren eines Programms
- Verwenden von Fehlerbehebungs-Tools ("Debugging")

Erforderliche Zeit für dieses Kapitel: 2 Stunden

Überblick

In diesem Kapitel erlernen Sie ausgewählte VEE-Programmiertechniken zum Erstellen eigener Programme. Mit VEE können Sie beispielsweise angepasste Objekte, die so genannten **UserObjects**, erstellen. Sie können auch Schnittstellen für die Benutzer (Operator) erstellen, in denen nur die erforderlichen Teile des Programms angezeigt werden. Diese Teile werden in der **Fensteransicht** des Programms angezeigt.

Sie können Daten von VEE aus in eine Datei schreiben und Daten aus einer Datei in VEE einlesen. Datendateien und die ihnen zugeordneten E/A-Transaktionen können für viele Bereiche angewendet werden einschließlich der Kommunikation mit Instrumenten, Dateien, Zeichenfolgen, dem Betriebssystem, den Schnittstellen, anderen Programmen, Rocky Mountain Basic und Druckern.

VEE unterstützt eine Vielzahl von Datentypen und bietet umfangreiche Funktionen zur mathematischen Verarbeitung von Daten. Sie haben verschiedene Möglichkeiten, VEE zur Kommunikation mit Instrumenten zu verwenden. VEE bietet darüber hinaus leistungsstarke Debugging-Tools zum Suchen und Beheben von Problemen in Programmen.

Allgemeine Techniken

Im VEE-Hauptprogramm können Sie logische Gruppen von Objekten, die so genannten `UserObjects`, erstellen. Ein `UserObject` (Benutzerobjekt) wird erstellt, indem Sie eine logische Gruppe in einem `UserObject`-Fenster platzieren. In diesem `UserObject`-Fenster verbinden Sie die Ein- und Ausgänge auf die gleiche Weise wie im Hauptprogramm. Das `UserObject` selbst ist mit anderen Objekten im Hauptprogramm mit Ein- und Ausgängen verbunden wie jedes andere Objekt auch.

Der Zweck bei der Entwicklung eines `UserObject` ist das Erstellen eines eindeutigen Kontext zur Ausführung einer bestimmten Aufgabe innerhalb des Hauptprogramms. Auf diese Weise sparen Sie nicht nur Platz im Hauptarbeitsbereich, sondern strukturieren das Programm und gestalten es dadurch übersichtlicher.

Ein VEE-Programm kann viele im Hauptprogramm verschachtelte `UserObjects` enthalten. Jedes `UserObject` enthält eine Symbolansicht im Hauptfenster. Wenn Sie die Symbolansicht der `UserObjects` im Hauptprogramm den entsprechenden `UserObject`-Fenstern zuordnen wollen, geben Sie den `UserObjects` in ihren Bearbeitungsfenstern einen Namen; dadurch werden sie auch in der entsprechenden Symbolansicht benannt. Wenn Sie beispielsweise ein `UserObject AddNoise` nennen, haben sowohl das Symbolfenster im Hauptprogramm als auch die Titelleiste im `UserObject` den Inhalt `AddNoise`. In der folgenden Übung lernen Sie, wie Sie ein `UserObject` erstellen.

Übung 2-1: Erstellen eines UserObject

Es gibt verschiedene Möglichkeiten, ein `UserObject` in einem VEE-Programm zu erstellen:

- Wählen Sie `Device` ⇒ `UserObject` in der Menüleiste aus, um ein leeres `UserObject`-Symbol im Hauptfenster (`Main`) anzuzeigen, und fügen Sie diesem UserObject Objekte hinzu. Wenn Sie doppelt auf das Symbol `UserObject` klicken, wird es in der offenen Ansicht angezeigt (siehe Abbildung 2-1).

- Wählen Sie Objekte in einem Programm aus und erstellen Sie anschließend ein `UserObject` daraus, indem Sie die Objekte auswählen und `Edit` ⇒ `Create UserObject` anklicken.

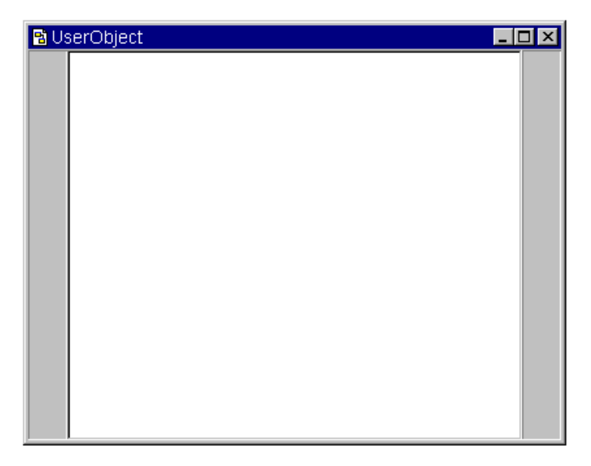


Abbildung 2-1. UserObject-Fenster

Nachdem Sie ein `UserObject` erstellt haben, ist es Teil des Hauptprogramms. Das `UserObject`-Fenster kann als Symbol angezeigt werden, in der offenen Ansicht oder in Symbolgröße am unteren Rand der Anzeige:

- Schließen Sie das Fenster durch Anklicken der Schaltfläche "Schließen", um das `UserObject` als Symbol im Hauptfenster anzuzeigen.
- Zur Anzeige des Fenster mit maximaler Größe klicken Sie seine Schaltfläche "Vollbild" an; das `UserObject`-Fenster belegt jetzt den gesamten verfügbaren Bereich im VEE-Arbeitsbereich.
- Stellen Sie das Fenster in Symbolgröße dar, indem Sie seine Schaltfläche "Symbolgröße" anklicken. Das `UserObject` erscheint jetzt in Symbolgröße am unteren Rand des VEE-Arbeitsbereichs.

Hinweis

Die Symbolansicht des `UserObject` befindet sich immer im Hauptfenster; in dieser Ansicht können Sie die Pins des Objekts mit anderen Objekten im Hauptfenster verbinden.

Hinweis

Stellen Sie zunächst sicher, dass der `Program Explorer` im Menü `View` nicht ausgewählt ist; Sie haben somit mehr Platz im Hauptfenster.

Als nächstes erstellen Sie ein `UserObject` für ein Programm.

1. Öffnen Sie das Programm (`simple-program.vee`), das Sie in “Hinzufügen eines Amplituden-Eingangs und eines Real64 Slider” auf Seite 74 erstellt haben. Das Programm sollte im Hauptarbeitsbereich angezeigt werden.

2. Entfernen Sie den `Real64 Slider` aus dem Programm (er wird in dieser Übung nicht verwendet). Klicken Sie, um das `Real64 Slider`-Objektmenü zu öffnen, und wählen Sie `Cut` aus, oder klicken Sie doppelt auf die Schaltfläche "Objektmenü" des `Real64 Slider`.

Hinweis

Wenn Sie das Programm jetzt erneut ausführen, nachdem das `Real64 Slider`-Objekt entfernt wurde und der Eingangs-Pin noch immer am `Noise Generator` angeschlossen ist, erhalten Sie eine VEE-Fehlermeldung, dass der Eingangs-Pin `Amplitude` am `Noise Generator` nicht angeschlossen ist. *Denken Sie daran, dass alle Eingangs-Pins angeschlossen sein müssen, damit ein VEE-Programm ausgeführt werden kann.*

3. Klicken Sie im Objekt `Noise Generator` die Schaltfläche **Objektmenü** an oder klicken Sie die rechte Schaltfläche über dem Objekt an, um das Objektmenü zu öffnen. Wählen Sie `Delete Terminal` ⇒ `Input` aus und klicken Sie im Dialogfenster für `Choose an input to delete` die Schaltfläche `OK` an, während `Amplitude` hervorgehoben ist.

4. Benennen Sie das Programm um, indem Sie `File` ⇒ `Save As...` auswählen und den neuen Namen `usrobj-program1.vee` eingeben.

5. Zeigen Sie anschließend das Objekt `Noise Generator` in Symbolgröße an, und ordnen Sie die Objekte wie in Abbildung 2-2 gezeigt neu an.

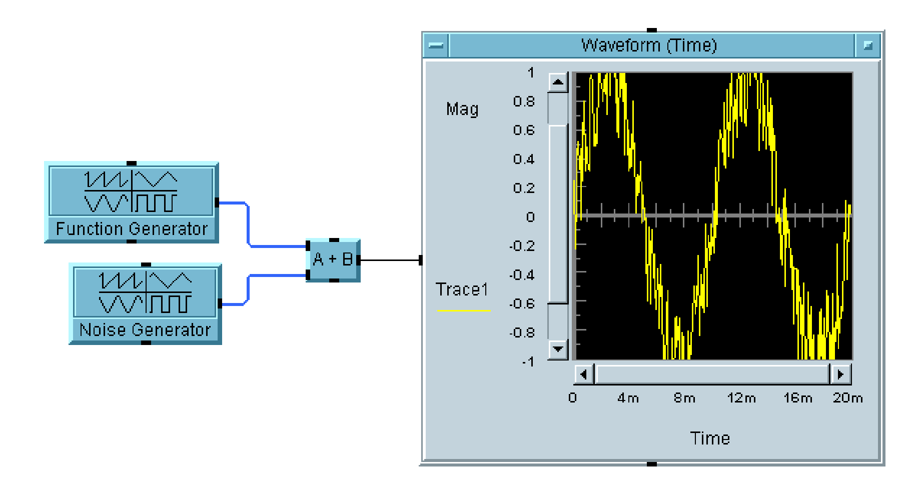


Abbildung 2-2. usrobj-program.vee ini einer frühen Phase

6. Wählen Sie die Objekte `Noise Generator` und `A+B` mit dem Direktaufruf **Strg**+*linke Maustaste* aus. Klicken Sie `Edit` ⇒ `Create UserObject` an. Es erscheint ein Dialogfenster mit der Bezeichnung `Create UserObject`. (Sie können das Objekt umbenennen, indem Sie einen neuen Namen eingeben. Für den Augenblick klicken Sie einfach `OK` an, um das `UserObject` zu erstellen.)

   Das `UserObject` enthält die Objekte `Noise Generator` und `A+B` im `UserObject`-Bearbeitungsfenster und wird automatisch im `Hauptfenster` mit den entsprechenden Ein- und Ausgangs-Pins erstellt, wie in Abbildung 2-3 dargestellt.

   Tipp: Positionieren Sie die Symbole in der linken oberen Ecke des `UserObject`, indem Sie einfach die Taste **Pos1** auf der Tastatur drücken.

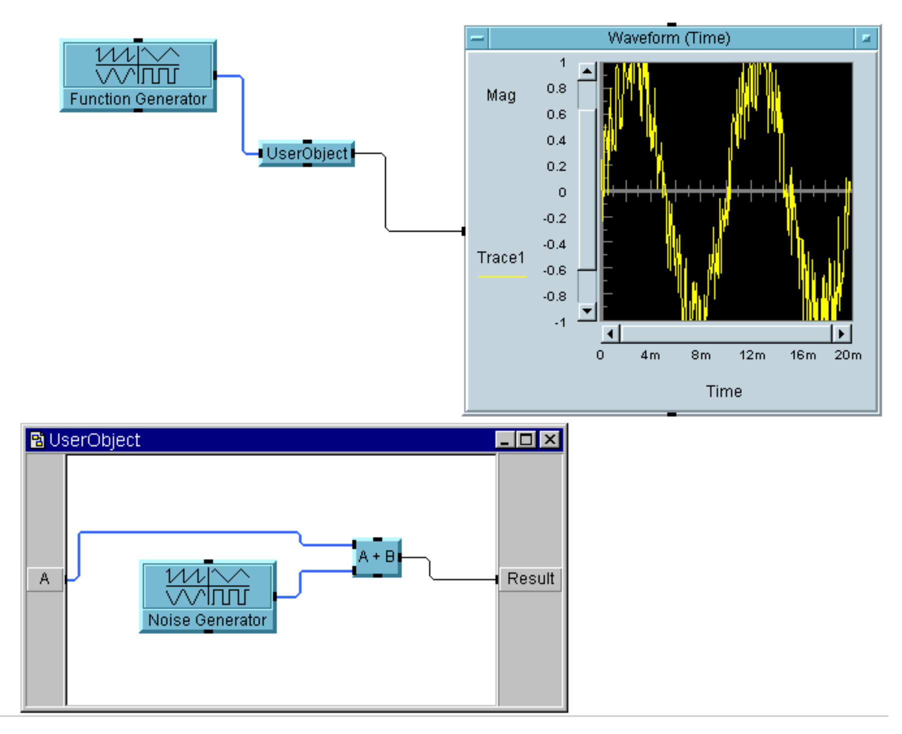


Abbildung 2-3. Erstellen eines UserObject

Hinweis

Das neu Anordnen der Objektpositionen vor der Ausführung von `Create UserObject` dient lediglich der besseren Übersichtlichkeit. Wenn Sie die in einen Bereich einzubeziehenden Objekte nicht erfassen, legt das `UserObject` seine Größe selbst fest, so dass es alle ausgewählten Objekte enthalten kann. Sie können den Arbeitsbereich des `UserObject` anschließend neu anordnen und seine Größe ändern sowie das `UserObject` an eine geeignete Stelle im Arbeitsbereich verschieben. Das Bereinigen ist jedoch einfacher, wenn Sie die Objekte bereits vorher logisch anordnen.

Hinweis

Sie können mit `Edit` ⇒ `Clean Up Lines` die Linienführung über ein Programm bereinigen. Dieser Befehl ist kontextabhängig. Zum Bereinigen der Linien für das `UserObject` muss dieses Fenster aktiv sein. Klicken Sie das `UserObject`-Fenster an und wählen Sie `Edit` ⇒ `Clean Up Lines` aus.

Tipp: Durch das Erstellen eines `UserObject` in seinem Bearbeitungsfenster und die anschließende Verwendung der Symbolansicht des `UserObject` können Sie auf dem Bildschirm Platz sparen.

7. Zur besseren Kontrolle des `UserObject` ändern Sie seinen Titel von `UserObject` in `AddNoise`. Klicken Sie doppelt auf die Titelleiste und geben Sie den neuen Titel im Dialogfenster 'Properties' ein. Abbildung 2-4 zeigt, wie das Programm dadurch übersichtlicher wird.

   Tipp: Zum schnellen Aufrufen des Dialogfensters 'Properties' eines Objekts klicken Sie doppelt auf seine Titelleiste.

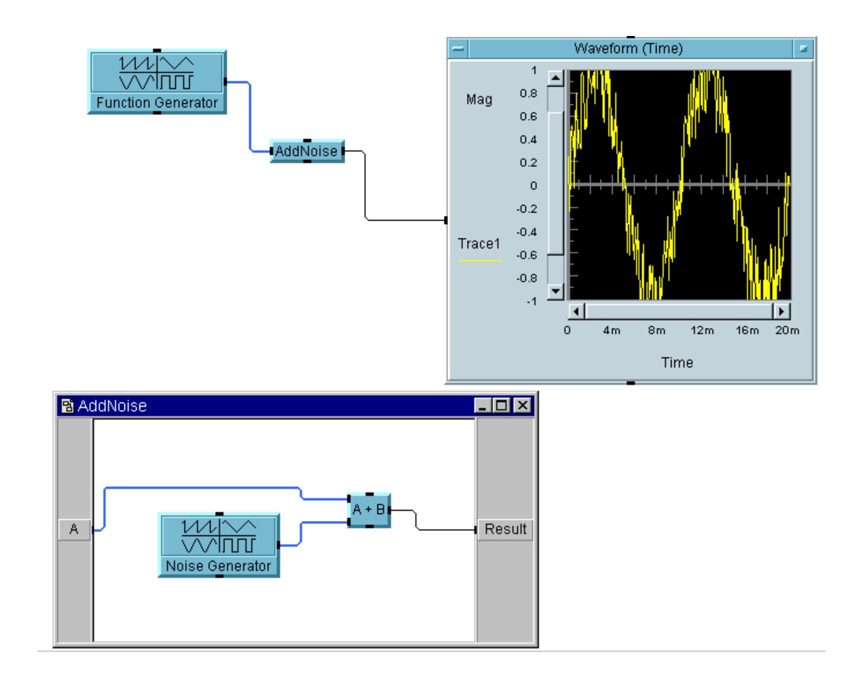


Abbildung 2-4. UserObject umbenannt in AddNoise

8. Klicken Sie die Schaltfläche **Ausführen** an, um die gestörte Kosinuskurve anzuzeigen, wie in Abbildung 2-5 dargestellt. Beachten Sie, dass `AddNoise` in Symbolgröße am unteren Rand des Arbeitsbereichs angezeigt wird. Zum Anzeigen von `AddNoise` in Symbolgröße klicken Sie die Schaltfläche 'Symbolgröße' in seiner Titelleiste an; diese Schaltfläche wird als Unterstreichungszeichen (_) dargestellt.

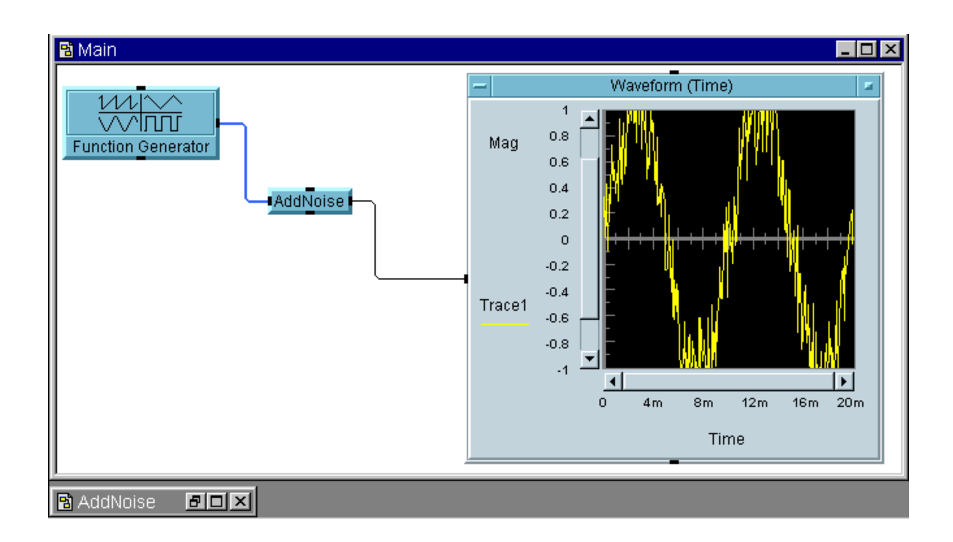


Abbildung 2-5. Gestörte Kosinuskurve

Der Schlüssel zu effizienten `UserObjects` ist das Sicherstellen, dass sie innerhalb des Programms einen logischen Zweck erfüllen. Dieses eindeutige Objekt spart nicht nur Platz, sondern ermöglicht auch ein effizientes Strukturieren eines Programms. `UserObjects` erleichtern die Verwendung eines “Top-Down”-Design in VEE-Programmen. VEE enthält auch ein Objekt `UserFunction`, ein mehrfach verwendbares Code-Modul. Weitere Informationen zu `UserObjects` und `UserFunctions` finden Sie in Kapitel 8, “Verwenden von Agilent VEE-Funktionen” auf Seite 309.

Weitere Informationen zu `UserObjects` können Sie durch Auswählen von `Help` ⇒ `Contents and Index` in der VEE-Menüleiste aufrufen. Blättern Sie anschließend durch `How Do I...`, `Tell Me About...` oder `Reference`.

Dieses Beispiel wird im folgenden Abschnitt fortgesetzt. Wenn Sie das Beispiel jedoch an dieser Stelle beenden wollen, speichern Sie das Programm unter dem Namen `usrobj-program3.vee`.

Übung 2-2: Erstellen eines Dialogfensters für die Benutzereingabe

Falls es nicht bereits geöffnet ist, öffnen Sie das Programm `usrobj-program3.vee`.

Im Untermenü `Data ⇒ Dialog Box` finden Sie sechs Auswahlfelder für Dialogfenster: `Text Input` (Texteingabe), `Int32 Input` (Int32-Eingabe) und `Real64 Input` (Real64 -Eingabe) sowie `Message Box` (Meldungsfeld), `List Box` (Listenfeld) und `File Name Selection` (Dateinamen-Auswahl). Bei der Eingabe von Text, Ganzzahlen (Integer) und realen Zahlen erleichtert ein entsprechendes Dialogfenster die Konfiguration der Eingabeaufforderung bzw. Beschriftung, der Standardwerte, Wertebegrenzungen und Fehlermeldungen. Wenn Sie eines dieser Dialogfenster verwenden, erscheint bei der Ausführung des Programms ein Einblendfenster.

1. Wählen Sie `Data ⇒ Dialog Box ⇒ Int32 Input` aus und platzieren Sie das Feld links vom `Function Generator`. Ändern Sie das Feld `Prompt/Label` in `Enter Frequency:`. (Sie müssen das Feld zunächst hervorheben, indem Sie darauf klicken und den Mauszeiger darüber ziehen.) Ändern Sie den `Default Value` (Standardwert) in `100`.

   Tipp: Sie können einen Eintrag auch hervorheben, indem Sie doppelt in das Eingabefeld klicken.

2. Ändern Sie die `Value Constraints` (Wertebegrenzungen) in `1` als unteren Wert und `193` als oberen Wert. Ändern Sie die Fehlermeldung entsprechend diesen neuen Werten, wie in Abbildung 2-6 dargestellt. Stellen Sie jetzt das Objekt `Int32 Input` als Symbol dar.

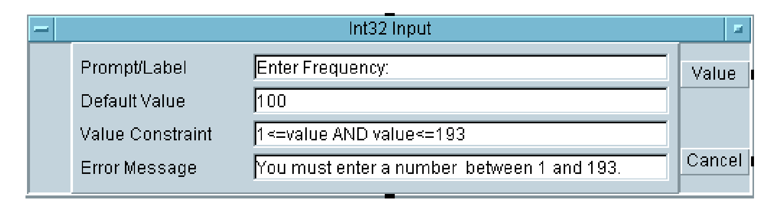


Abbildung 2-6. Das Konfigurationsfeld "Int32 Input"

3. Öffnen Sie das Objektmenü für den Function Generator und wählen Sie Add Terminal ⇒ Data Input aus. Wählen Sie im Dialogfenster für Select input to add die Option Frequency aus und klicken Sie OK an.

4. Verbinden Sie den oberen Ausgangs-Pin des Objekts Int32 Input mit dem Eingangs-Pin des Function Generator. Beachten Sie, dass Frequency jetzt nur über den Eingangs-Pin geändert werden kann; Sie können das Eingabefeld Frequency nicht mehr ändern. Das Programm sollte jetzt aussehen wie in Abbildung 2-7.

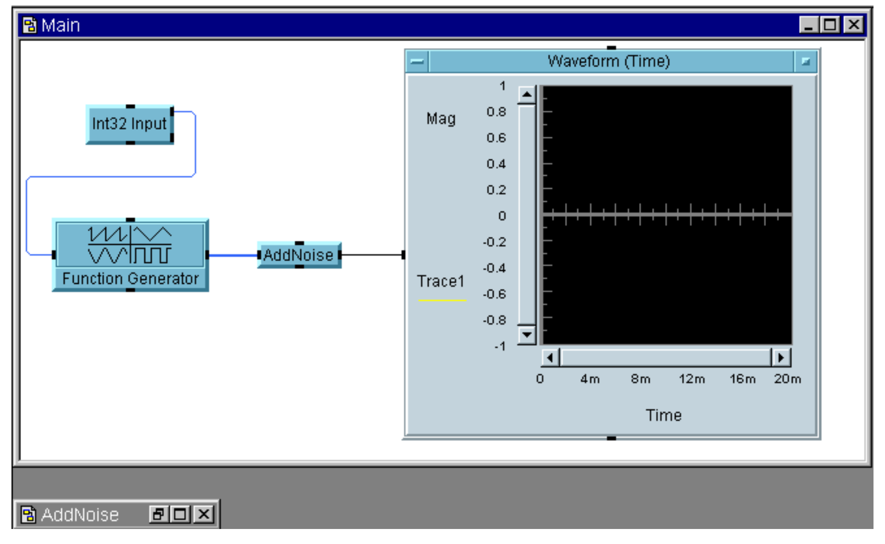


Abbildung 2-7. Int32-Eingabe zu usrobj-program.vee hinzugefügt

5. Führen Sie das Programm aus. Das Eingabefenster für Int32 Input wird angezeigt mit der Aufforderung Enter Frequency:. Führen Sie das Programm mit verschiedenen Frequenzen im Eingabefenster aus. In Abbildung 2-8 ist die Anzeige zur Ausführungszeit mit dem Einblend-’Eingabefenster dargestellt. Sie können das Einblendfeld anklicken und ziehen, um seine Position auf dem Bildschirm festzulegen.

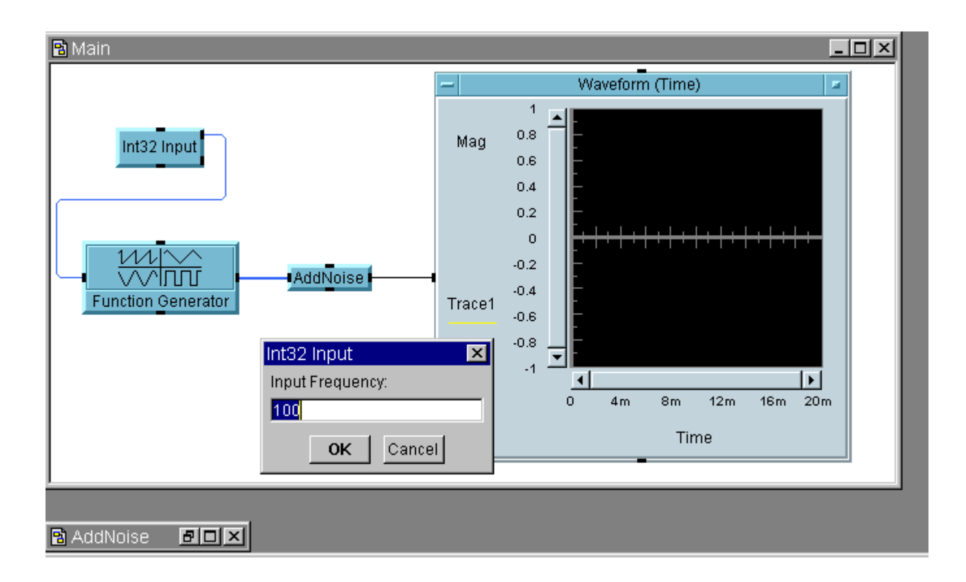


Abbildung 2-8. Einblend-Eingabefenster zur Ausführungszeit

Wenn Sie eine Frequenz über 193 eingeben, wird ein Fenster mit einer Fehlermeldung eingeblendet. Diese Meldung zeigt genau den von Ihnen eingegebenen Text an.

Dieses Beispiel wird im folgenden Abschnitt fortgesetzt. Wenn Sie das Beispiel jedoch an dieser Stelle beenden wollen, speichern Sie das Programm unter dem Namen `usrobj1-program4.vee`.

Hinweis

Die VEE -Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter `Help` ⇒ `Open Example...` ⇒ `Manual` ⇒ `UsersGuide` enthalten.

Übung 2-3: Verwenden von Datendateien

Sie können Daten aus VEE in eine Datendatei schreiben und Daten aus einer Datei in VEE einlesen. Sie verwenden hierzu die Objekte `To File` und `From File` im Programm. Fügen Sie beispielsweise ein Objekt `To File` der Detailansicht des von Ihnen erstellten Programms hinzu.

Falls es nicht bereits geöffnet ist, öffnen Sie das Programm `usrobj-program4.vee`.

1. Wählen Sie `I/O` ⇒ `To` ⇒ `File` aus und platzieren Sie dieses Objekt im Hauptarbeitsbereich.

2. Ändern Sie den standardmäßigen Dateinamen `myFile` in `wavedata`.

   Falls kein Häkchen links von `Clear File At PreRun & Open` eingetragen ist, klicken Sie auf das kleine Eingabefenster. `To File` hängt standardmäßig Daten an die vorhandene Datei an. In diesem Fall wollen Sie jedoch die Datei bei jeder Ausführung des Programms löschen. Das Objekt `To File` sollte jetzt aussehen wie in Abbildung 2-9.

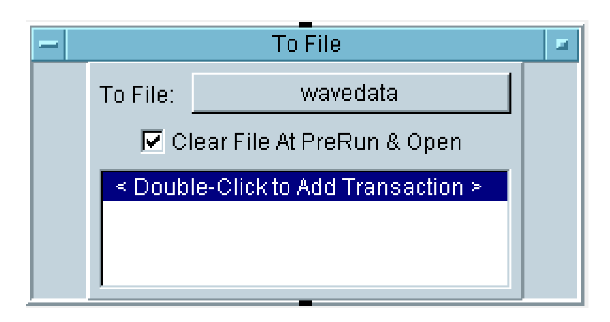


Abbildung 2-9. Hinzufügen einer Datendatei

3. Klicken Sie doppelt auf den Bereich mit der Beschriftung `Double-Click to Add Transaction`, um die Daten zu schreiben. Das in Abbildung 2-10 dargestellte Dialogfenster wird angezeigt. Klicken Sie das Feld `TEXT` (oder seinen Pfeil) an, um die Dropdown-Liste der Datentypen anzuzeigen, und klicken Sie `CONTAINER` an. Klicken Sie `OK` an. Wenn Sie `OK` im Dialogfenster `I/O Transaction` anklicken, wird automatisch ein Eingangs-Pin `a` dem Objekt `To File` hinzugefügt.

   Überprüfen Sie `Help` im `To File`-Objektmenü, um die weiteren Optionen für Transaktionen außer `WRITE CONTAINER` anzuzeigen. Transaktionen werden ausführlich beschrieben im Anhang des Handbuchs *VEE Pro Advanced Techniques* und in Kapitel 5, “Speichern und Abrufen von Testergebnissen”

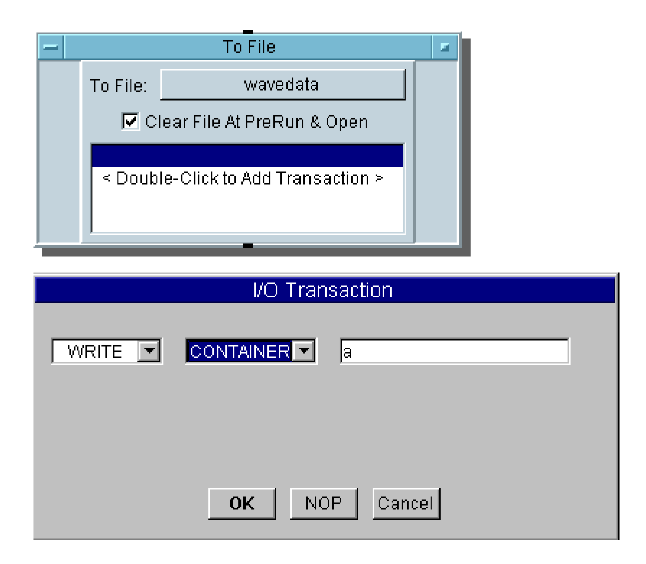


Abbildung 2-10. Auswählen einer E/A-Transaktion

4. Verbinden Sie den Datenausgangs-Pin von `AddNoise` mit dem Dateneingangs-Pin von `To File`. Das Programm sollte jetzt aussehen wie in Abbildung 2-11.



Hinweis

Sie können einen Datenausgangs-Pin mit mehreren Dateneingangs-Pins verbinden.

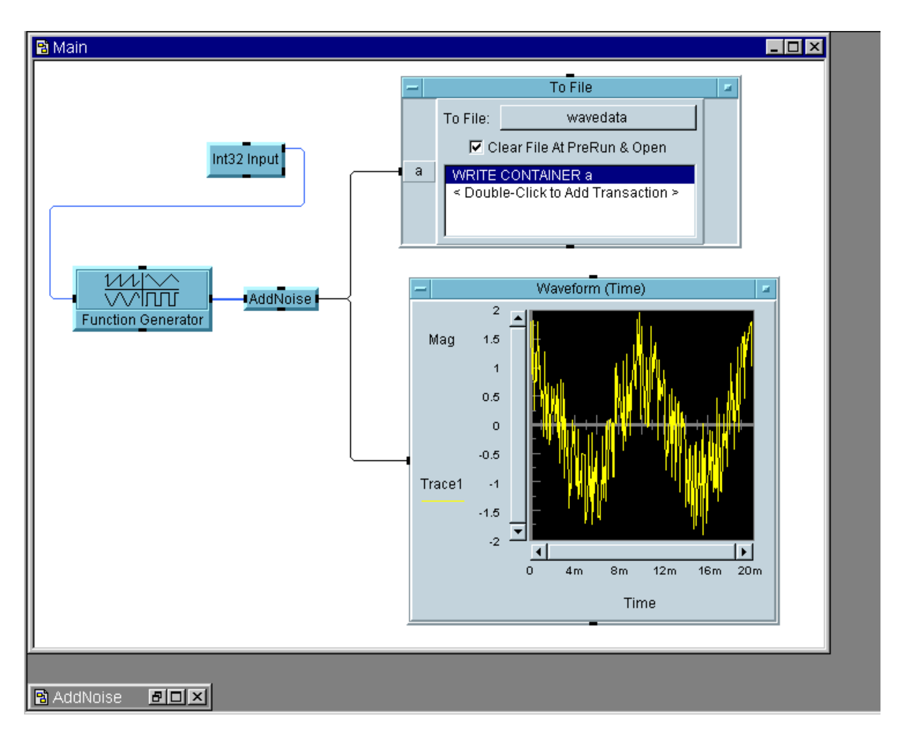


Abbildung 2-11. Hinzufügen eines To File-Objekts

5. Klicken Sie die Schaltfläche **Ausführen** in der Symbolleiste erneut an, um das Programm zu testen. Das Programm zeigt jetzt die durch das UserObject `AddNoise` gestörte Kosinuskurve an und schreibt einen Container mit Wellenformdaten in die Datei `wavedata`.

   Klicken Sie doppelt auf das Objekt `To File`, um die offene Ansicht aufzurufen, und klicken Sie anschließend doppelt auf den Eingangsanschluss `a`, um seinen Inhalt zu überprüfen. Es wird ein Array mit 256 Punkten angezeigt.

Fügen Sie dem Programm ein Objekt `From File` hinzu, um die Daten wieder einzulesen.

6. Wählen Sie `I/O` ⇒ `From` ⇒ `File` aus und platzieren Sie dieses Objekt im Hauptarbeitsbereich. Fügen Sie dem `READ CONTAINER x` eine Lese-Transaktion hinzu und ändern Sie den Dateinamen in

wavedata (die Prozedur ist die gleiche wie für To File). Löschen Sie anschließend die Linie zwischen AddNoise und dem Objekt Waveform Time), und verbinden Sie die Objekte wie in Abbildung 2-12 gezeigt. Die Sequenzlinie zwischen To File und From File sorgt dafür, dass die Daten in die Datei geschrieben werden, bevor sie gelesen werden.

7. Führen Sie das Programm aus. Es sollte ähnlich aussehen wie in Abbildung 2-12. Speichern Sie das Programm als usrobj-program.vee.

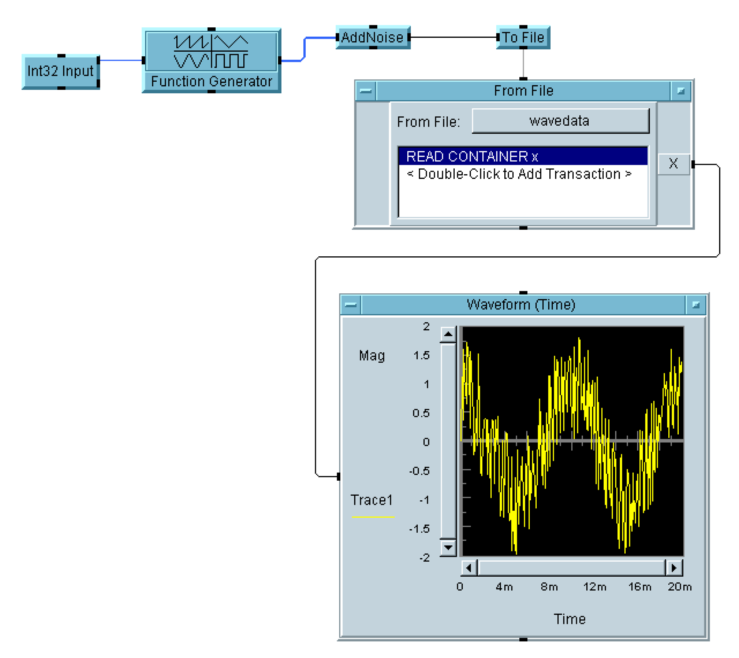


Abbildung 2-12. Hinzufügen eines From File-Objekts

Übung 2-4: Erstellen einer Fensteransicht (Bedienerschnittstelle)

Nach dem Entwickeln eines Programms können Sie eine Bedienerschnittstelle erstellen. Erstellen Sie hierzu eine **Fensteransicht** des Programms. In dieser Übung wird das Programm verwendet, das Sie in “Anzeige von Datenfluss und Weitergabe” auf Seite 71 erstellt haben.

1. Öffnen Sie das Programm `simple-program.vee`. Das Programm sollte aussehen wie in Abbildung 2-13.

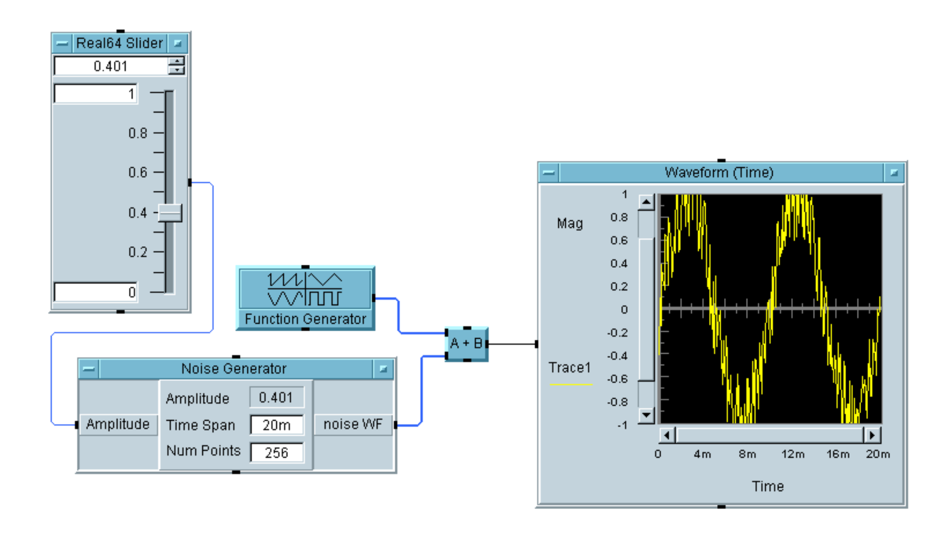


Abbildung 2-13. simple-program.vee

2. Wählen Sie die Objekte aus, die in der `Fensteransicht` erscheinen sollen. Die Fensteransicht dient als Bedienerschnittstelle. Drücken Sie die Taste **Strg** und halten Sie sie gedrückt, während Sie nacheinander die gewünschten Objekte anklicken. (Vergewissern Sie sich, dass kein Objekt versehentlich ausgewählt wird.) Wählen Sie in diesem Fall die Objekte `Real64 Slider` und `Waveform (Time)` aus. Diese beiden Objekte haben jeweils einen Schatten zur Kennzeichnung, dass sie ausgewählt wurden.

3. Klicken Sie die Schaltfläche `Zum Fenster hinzufügen` in der Symbolleiste an, um die ausgewählten Objekte dem Fenster hinzufügen (oder verwenden Sie `Edit` ⇒ `Add To Panel`). Es erscheint eine Fensteransicht mit den beiden Objekten, die Sie dem Fenster hinzugefügt haben.

   Sie können die Objekte in der Fensteransicht an die gewünschte Position verschieben und ihre Größe ändern, um eine Anzeige wie die in Abbildung 2-14 zu erhalten.

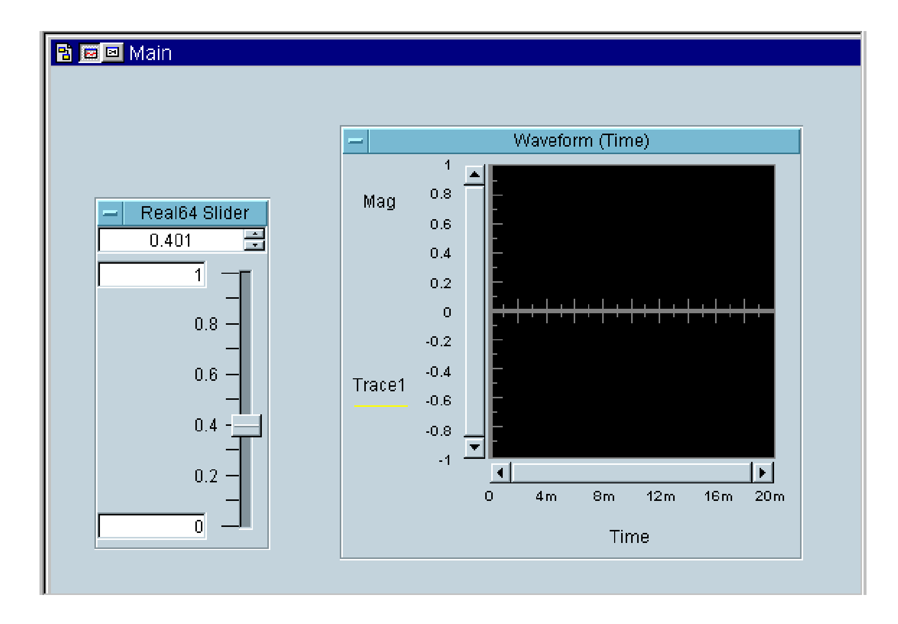


Abbildung 2-14. Beispiel: Erstellen einer Fensteransicht

4. Drücken Sie die Schaltfläche **To Detail** links oben in der Titelleiste des `Hauptfensters`, um zur Detailansicht zu wechseln. Klicken Sie die Schaltfläche **To Panel** an, um zur Fensteransicht zurückzukehren.

   Die Detailansicht entspricht dem normalen Fenster, in dem Sie ein Programm bearbeiten. Sie können Objekte in der Fensteransicht unabhängig von der Detailansicht verschieben, löschen oder ihre Größe ändern. Die Detailansicht wird zum Entwickeln eines Programms verwendet; die Fensteransicht dagegen zum Bereitstellen einer Bedienerschnittstelle.

5. Speichern Sie das Programm als `simple-program_with_panel.vee`.

Sie können mit der Fensteransicht üben, indem Sie die folgenden Änderungen daran vornehmen:

■ Wählen Sie zum Ändern der Farben in der Ansicht `Properties` im Objektmenü des Hauptfensters in der Fensteransicht aus. Wählen Sie anschließend `Colors` aus, klicken Sie die Schaltfläche `Fensteransicht` ⇒ `Background:` an und wählen Sie die gewünschte Farbe aus.

- Klicken Sie zum Ändern der Farben oder Schriften doppelt auf den Titel eines beliebigen Objekts, um das Feld `Properties` aufzurufen. Klicken Sie anschließend das Register `Colors` oder `Fonts` an und nehmen Sie die gewünschten Änderungen vor.

- Wenn die Objekte in der Fensteransicht leicht abgehoben dargestellt werden sollen, öffnen Sie das Fenster `Properties` für das entsprechende Objekt, öffnen Sie den Ordner `Appearance` durch Anklicken des entsprechenden Registers, und wählen Sie `Raised` unter `Border` aus.

- Zum Ändern des Namens der Fensteransicht öffnen Sie das Dialogfenster `Properties`, und geben Sie der Fensteransicht einen Namen Ihrer Wahl. Der eingegebene Name wird bei der Ausführung des Programms angezeigt.

Übung 2-5: Mathematische Verarbeitung von Daten

VEE bietet umfangreiche integrierte mathematische Funktionen und unterstützt eine Vielzahl mathematischer Datentypen wie auch die Daten- und Signalverarbeitung von MATLAB. Ausführliche Hinweise hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Verwenden von Datentypen

VEE unterstützt verschiedene Datentypen einschließlich Text, Integer (Ganzzahlen) und Real (reale Zahlen) sowie verschiedene Arten von komplexen und Koordinatenzahlen. Sie haben bereits in früheren Beispielen gesehen, wie das Objekt `A+B` zwei Wellenformen addieren kann. Mathematische Operatoren wie Addition `(+)` können mit verschiedenen Datentypen und sogar mit Mischtypen verwendet werden.

Löschen Sie zum Erstellen des folgenden Beispielprogramms zunächst den Inhalt des `Hauptfensters`. Platzieren Sie jetzt die folgenden Objekte im `Hauptfenster` und verbinden Sie sie wie gezeigt; notieren Sie dabei die folgenden Informationen.

1. Wählen Sie `File` ⇒ `New` aus, um den Arbeitsbereich zu löschen.

2. Fügen Sie ein Objekt `Real64 Constant` hinzu, indem Sie `Data` ⇒ `Constant` ⇒ `Real64` auswählen.

3. Fügen Sie ein Objekt `Complex Constant` hinzu, indem Sie `Data` ⇒ `Constant` ⇒ `Complex` auswählen.

4. Fügen Sie ein Objekt `A+B` hinzu. Wählen Sie `Device` ⇒ `Function & Object Browser` aus, um den `Function & Object Browser` aufzurufen. Wählen Sie anschließend `Type: Operators`; `Category: Arithmetic`; `Operators: +` aus. Klicken Sie `Create Formula` an, um das Objekt zu erstellen.

5. Fügen Sie ein Objekt `AlphaNumeric` hinzu, indem Sie `Display` ⇒ `AlphaNumeric` auswählen. Verbinden Sie die Objekte wie in Abbildung 2-15 gezeigt. Geben Sie den Wert `1.53` im Dateneingabefeld des Objekts `Real64 Constant` und den komplexen Wert `(2,1)` im Objekt `Complex` ein. Führen Sie das Programm aus. Das Ergebnis sollte wie das in Abbildung 2-15 aussehen.

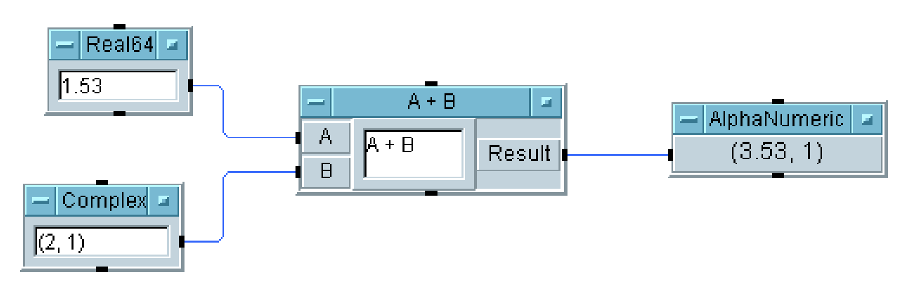


Abbildung 2-15. Verwenden von Datentypen

VEE wandelt die Daten automatisch nach Bedarf um und führt die Addition in dem Objekt `A+B` durch. Der reale Wert `1.53` wird in den komplexen Wert `(1.53,0)` umgewandelt und anschließend zu dem komplexen Wert `(2,1)` addiert. Das Ergebnis `(3.53,1)` (eine komplexe Zahl) wird im Objekt `AlphaNumeric` angezeigt.

Hinweis

Normalerweise verarbeitet VEE alle Umwandlungen von Datentypen automatisch. Weitere Informationen können Sie über `Help` ⇒ `Contents and Index` in der VEE-Menüleiste aufrufen. Blättern Sie anschließend durch `How Do I...`, `Tell Me About...` oder `Reference`.

Verwenden von Datenformen

VEE unterstützt eine Vielzahl von Datenformen wie beispielsweise Skalare und Arrays. Anders als bei den meisten anderen Programmiersprachen können VEE-Objekte mit einem vollständigen Array verwendet werden und nicht nur mit einzelnen Elementen.

Das folgende Programm erstellt einen eindimensionalen Array aus zehn Elementen, berechnet den Mittelwert der 10 Werte und zeigt diesen Mittelwert an.

1. Wählen Sie `File` ⇒ `New` aus, um den Arbeitsbereich zu löschen.

2. Fügen Sie ein Objekt `For Range` hinzu, indem Sie `Flow` ⇒ `Repeat` ⇒ `For Range` auswählen.

3. Fügen Sie ein Objekt `Sliding Collector` hinzu, indem Sie `Data` ⇒ `Sliding Collector auswählen.`

4. Fügen Sie ein Objekt `median(x)` hinzu. Wählen Sie `Device` ⇒ `Function & Object Browser` aus. Wählen Sie anschließend `Type: Built-in Functions`; `Category: Probability & Statistics`; `Functions: median` aus, und klicken Sie `Create Formula` an.

Shortcut: Sie können den `Function & Object Browser` anzeigen, indem Sie die Schaltfläche ***fx*** in der Symbolleiste auswählen.

5. Fügen Sie ein Objekt `AlphaNumeric` hinzu, indem Sie `Display` ⇒ `AlphaNumeric` auswählen. Verbinden Sie die Objekte wie in Abbildung 2-16 gezeigt. Führen Sie das Programm aus. Wenn Sie keine der Eingänge an den Objekten geändert haben, sollte das in Abbildung 2-16 dargestellte Ergebnis angezeigt werden.

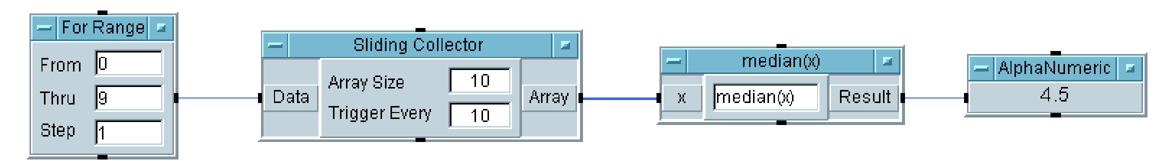


Abbildung 2-16. Verbinden von Datenobjekten

Verwenden des Formelobjekts

VEE bietet mathematische Operatoren und Funktionen, die im Abschnitt Reference der Online-Hilfe beschrieben sind. Wählen Sie Help ⇒ Contents and Index aus. Wählen Sie anschließend Reference aus und blättern Sie durch die angezeigten Elemente.

Die vordefinierten Operator- und Funktionsobjekte sind über Device ⇒ Function & Object Browser (oder ***fx*** in der Symbolleiste) verfügbar. Sie wählen sie über den Function & Object Browser aus, indem Sie die Einheiten in den drei Listen anklicken: Type:, Category: und Functions:. Klicken Sie Create Formula an, um das Objekt zu erstellen.

Neben den vordefinierten Operatoren und Funktionen können Sie jeden beliebigen gültigen mathematischen VEE-Ausdruck mit dem Formula-Objekt erstellen; Sie finden dieses Objekt im Menü Device. In diesem Abschnitt erstellen Sie ein Programm mit einem Formula-Objekt. Löschen Sie zunächst den Inhalt des Hauptfensters, und gehen Sie anhand der folgenden Schritte vor:

1. Fügen Sie das Objekt Function Generator dem Hauptfenster hinzu und ändern Sie es, sodass sich eine 100 Hz-Sinuskurve ergibt. Wählen Sie Device ⇒ Virtual Source ⇒ Function Generator aus.

2. Wählen Sie Device ⇒ Formula aus, um das Formula-Objekt dem Hauptfenster hinzuzufügen. Fügen Sie dem Objekt einen zweiten Eingang (B) hinzu, indem Sie mit dem Mauszeiger in den Eingangsanschlussbereich zeigen und **Strg**+**A** drücken.

3. Geben Sie den mathematischen Ausdruck abs(A)+B im Eingabefeld ein.

4. Wählen Sie Data ⇒ Constant ⇒ Real64 aus, um ein Objekt Real64 Constant dem Hauptfenster hinzuzufügen. Geben Sie den Wert 0.5 ein.

5. Wählen Sie Display ⇒ Waveform (Time) aus und legen Sie die Skala der y-Achse (y-axis) auf -2 bis 2 fest. Stellen Sie Automatic Scaling (Automatische Skalierung) auf Off (Aus). Klicken Sie zum Aufrufen des Dialogfensters für diese Parameter Mag an.

6. Verbinden Sie die Objekte wie in Abbildung 2-17 gezeigt. Führen Sie das Programm aus.

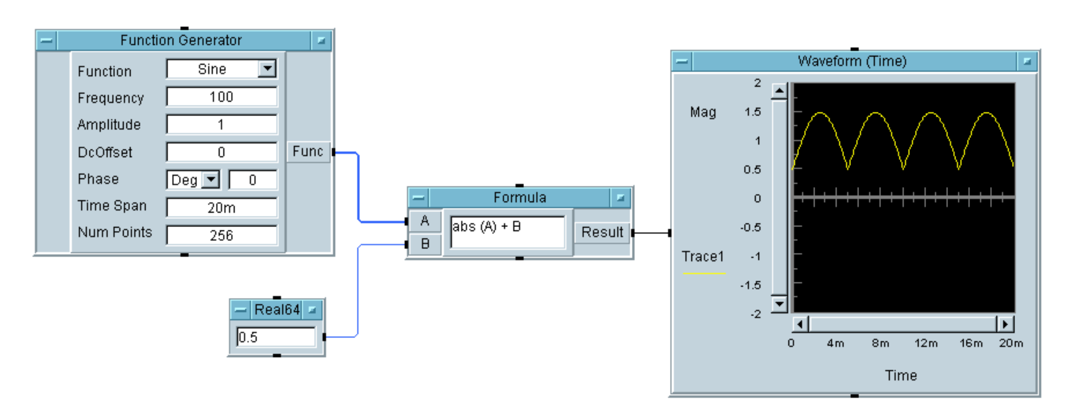


Abbildung 2-17. Erstellen eines Programms mit Formelobjekt

Bei der Ausführung des Programms verwendet das `Formula`-Objekt den Wellenformeingang `A` sowie den realen Wert `B` und addiert `B` zu dem absoluten Wert von `A`. Der Ausdruck `abs(A)+B` führt ein "Gleichrichten" der Sinuskurve durch und addiert einen "Gleichstromversatz". Sie können das gleiche Ergebnis auch über die Objekte `A+B` und `abs(x)` erzielen, es ist jedoch einfacher, einen Ausdruck in ein `Formula`-Objekt einzulesen. (Außerdem sparen Sie dadurch Platz.)

Klicken Sie doppelt auf die Ein- und Ausgangsanschlüsse des `Formula`-Objekts. Beachten Sie, dass das reale Skalar an Eingang `B` jedem Element der Wellenformdaten (einem eindimensionalen Array) an Eingabe `A` hinzugefügt wird; die resultierende Wellenform ist der Ausgang am Anschluss `Result`.

Hinweis

Zur Ergänzung der umfangreichen mathematischen Funktionen von VEE stehen über die MATLAB Script-Integration Hunderte weiterer mathematischer Funktionen zur Verfügung. Sie können im `Function & Object Browser` durch diese Funktionen blättern. Weitere Informationen zu MATLAB-Funktionen finden Sie im Abschnitt “Verwenden von MATLAB Script in Agilent VEE” auf Seite 194 in Kapitel 4, “Analysieren und Anzeigen von Testdaten”.

Verwenden der Online-Hilfe

Sie haben jetzt einige einfache Programme erstellt - jetzt werden Sie einige Möglichkeiten kennen lernen, wie Sie mehr über VEE erfahren können.

1. Führen Sie zunächst die Multimedia-Lernprogramme im Menü `Help` ⇒ `Welcome` aus. Die Lernprogramme demonstrieren viele der wichtigsten Funktionen von VEE. Sie werden ihnen helfen, in kurzer Zeit effizient arbeiten zu können. Die Lernprogramme zeigen Bildschirmdemonstrationen zum Erstellen und Ausführen von VEE-Programmen an mit Erläuterungen zu der jeweiligen Anzeige. In den Lernprogrammen werden auch Konzepte für einen effizienten Einsatz von VEE vorgestellt.

2. Wenn Sie mit VEE vertraut sind, schlagen Sie weitere Informationen in den `Help`-Einträgen der Objektmenüs nach. Sie können mit den Objekten experimentieren, bis Sie ihre Arbeitsweise vollständig verstanden haben. Wenn Sie mehr über ein Objekt erfahren wollen, finden Sie spezifische Informationen in den jeweiligen Objektmenüs. Schlagen Sie zunächst in diesen Menüs nach.

3. Wenn Sie mit den Hilfethemen, dem Index oder den Suchfunktionen arbeiten wollen, rufen Sie `Help` in der VEE-Hauptmenüleiste auf.

Hinweis

Eine Übersicht zum Öffnen der `Hilfe`-Hauptfunktion und zum Öffnen einer Liste der `Hilfe`-Inhalte finden Sie im Abschnitt “Hilfe aufrufen” auf Seite 26 in Kapitel 1, “Verwenden der Agilent VEE-Entwicklungsumgebung”.

Verwenden der Hilfefunktion

Die Online-Hilfe bietet Informationen zu folgenden Themen:

- Alle Menüelemente sowie die meisten Direktaufrufe dazu
- Informationen zu Gerätetreibern
- Häufig ausgeführte Aufgaben und viele Beispielprogramme
- Definition der VEE-Begriffe

- Verwenden der Hilfefunktion
- VEE-Version

Sie können durch die Themen blättern, den Schlüsselwort-Index verwenden, Hyperlinks zu verwandten Themen aufrufen oder sogar Suchfunktionen verwenden. VEE umfasst viele Hilfefunktionen, die Sie bei der Entwicklung von Programmen einsetzen können.

Hinweis

Außerdem enthält VEE weitere hilfreiche Funktionen für die Entwicklung und Fehlerbehebung von Programmen wie beispielsweise "Line Probe". Weitere Informationen hierzu finden Sie im Abschnitt “Fehlerbehebung bei Programmen in Agilent VEE” auf Seite 106.

Anzeigen von Hilfe zu einem Objekt

Zum Aufrufen von Hilfe zu einem Objekt klicken Sie die Schaltfläche "Objektmenü" an und wählen `Help` aus.

- Wählen Sie `Flow` ⇒ `Repeat` ⇒ `For Count` aus, um ein Objekt `For Count` zu erstellen. Klicken Sie das Objektmenü an und wählen Sie `Help` aus. Das `Hilfethema` mit der Beschreibung zu `For Count` wird angezeigt.
- Wählen Sie `Device` ⇒ `Formula` aus, um ein Objekt `Formula` zu erstellen. Klicken Sie das Objektmenü an und wählen Sie `Help` aus. Das `Hilfethema` erscheint mit einer Beschreibung der im `Formula`-Objekt angezeigten Formel.
- Wählen Sie `Device` ⇒ `Function & Object Browser` aus. Wählen Sie eine Kombination der Auswahloptionen aus und klicken Sie `Help` an. Das `Hilfethema` zu dem ausgewählten Objekt wird angezeigt.

Suchen der Menüposition eines Objekts

Zum Suchen der Position eines Objekts in den Menüs und zum Anzeigen der Informationen zu diesem Objekt wählen Sie `Help ⇒ Contents and Index` aus, klicken das Register `Index` an, geben den Namen des Objekts ein und klicken `Display` an.

Wählen Sie beispielsweise `Help ⇒ Contents and Index` aus, klicken Sie das Register `Index` an und geben Sie `Collector` ein. Klicken Sie `Display` an, um das Hilfethema zu dem Objekt `Collector` anzuzeigen.

Weitere praktische Übungen mit der Hilfefunktion

- Nachschlagen des Direktaufrufs zum Löschen eines Objekts.

  Wählen Sie `Help ⇒ Contents and Index ⇒ How Do I... ⇒ Use the Keyboard Shortcuts ⇒ Editing Programs ⇒ To Cut an Object or Text` aus.

- Nachschlagen des Stichworts “terminal” (Anschluss).

  Wählen Sie `Help ⇒ Contents ⇒ Reference ⇒ Glossary ⇒ Terminal` aus.

- Nachschlagen der VEE-Versionsnummer.

  Wählen Sie `Help ⇒ About VEE Pro` aus.

- Feststellen, was in dieser Version von Agilent VEE neu ist.

  Wählen Sie `Help ⇒ Contents and Index ⇒ What’s New in Agilent VEE 6.0` aus.

Fehlerbehebung bei Programmen in Agilent VEE

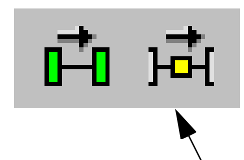
In dieser Übung wird das Programm verwendet, das Sie in “Erstellen einer Fensteransicht (Bedienerschnittstelle)” auf Seite 2-4 erstellt haben. Wählen Sie `File ⇒ Open` aus, heben Sie `simple-program_with_panel.vee` hervor und klicken Sie `OK` an.

VEE zeigt während der Entwicklung und bei der Ausführung eines Programms Fehlermeldungen an; außerdem können Meldungen des Typs "Caution" (Achtung), "Error" (Fehler) oder "Information" wie folgt angezeigt werden:

- Bei der Ausführung eines Programms zeigt VEE eventuell ein Fenster mit der gelben Überschrift `Caution` an.
- Bei der Ausführung eines Programms zeigt VEE eventuell ein Fenster mit der roten Überschrift `Error` an.
- Wenn Sie beim Erstellen eines Programms einen Fehler gemacht haben, beispielsweise die Eingabe eines Werts außerhalb des zulässigen Bereichs (`33000` in einem `Int16 Constant`-Feld), zeigt VEE ein `Error`-Meldungsfenster mit einer dunkelblauen Titelleiste an.
- Außerdem zeigt VEE Informationen zu Fehlern und Caution-Meldungen in der Statusleiste an. Die Statusleiste ist die Leiste um unteren Rand des VEE-Fensters.

Anzeigen des Datenflusses

1. Klicken Sie die Schaltfläche **Datenfluss anzeigen** in der Mitte der Symbolleiste an, wie in Abbildung 2-18 gezeigt. (Sie können stattdessen auch `Debug ⇒ Show Data Flow` anklicken.)



Schaltfläche "Datenfluss anzeigen" in der Symbolleiste

Abbildung 2-18. Datenfluss anzeigen

(Zum Ausschalten klicken Sie die Schaltfläche erneut an.) Bei der Ausführung des Programms werden kleine Quadrate entlang der Datenlinien verschoben, die den Datenfluss kennzeichnen.

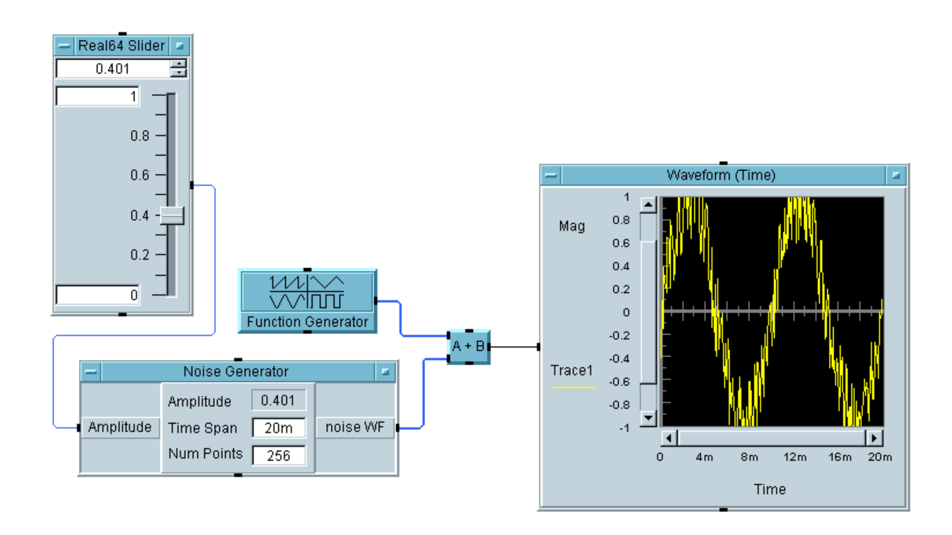
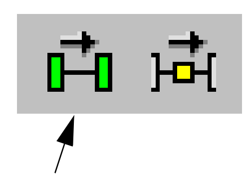


Abbildung 2-19. Datenfluss in simple-program.vee

In Abbildung 2-19 werden beispielsweise Daten von `Real64 Slider` zum Objekt `Noise Generator` verschoben. Der Ausgang vom `Noise Generator` und dem `Function Generator` wird als Eingang für das Objekt `A+B` verwendet, und die Ergebnisse erscheinen in der Anzeige `Waveform (Time)`.

Anzeigen des Ausführungsflusses

1. Klicken Sie die Schaltfläche **Ausführungsfluss anzeigen** in der Symbolleiste an, wie in Abbildung 2-20 gezeigt. (Sie können auch `Debug` ⇒ `Show Execution Flow` anklicken.)



Schaltfläche "Ausführungsfluss anzeigen" in der Symbolleiste

Abbildung 2-20. Ausführungsfluss anzeigen

Bei der Ausführung eines Programms wird eine farbige Umrisslinie um die gerade ausgeführten Objekte angezeigt.

Verwenden Sie `Datenfluss` und `Ausführungsfluss`, wenn Sie die Arbeitsweise eines Programms nachvollziehen wollen. Für eine höhere Leistung sollten Sie diese Funktionen jedoch ausschalten. Durch Kombinieren dieser Funktionen mit Fehlerbehebungs-Tools wie **Unterbrechungspunkten** können Sie leichter nachvollziehen, wie ein VEE-Programm arbeitet und wo mögliche Fehlerquellen liegen.

Untersuchen von Daten an einer Linie

Das Überprüfen von Daten an verschiedenen Stellen in Ihrem Programm ist eine schnelle und hilfreiche Methode, Fehler in Ihrem Programm zu finden und zu beheben. Die **Line Probe** (Linienprüfung) ist eine Möglichkeit, die Daten an einer gegebenen Linie zu überprüfen.

Positionieren Sie den Mauszeiger auf einer Datenlinie in der Detailansicht. Der Cursor wird als grafische Darstellung eines Vergrößerungsglases angezeigt. Die Linie und ihre Verbindungen werden hervorgehoben, und es erscheint ein Feld mit dem Datenwert an der Linie. Klicken Sie den Cursor (das Vergrößerungsglas) an, um ein Dialogfenster mit weiteren Informationen zu der Datenlinie anzuzeigen. (Sie können auch `Debug` ⇒ `Line Probe` und anschließend eine Linie anklicken.)

Abbildung 2-21 zeigt beispielsweise ein VEE-Programm mit dem Ausgang von dem als Symbol angezeigten `Function Generator` an. Die Ausgabe zeigt, dass der `Function Generator` einen Wellenform-Array mit 256 Punkten generiert.

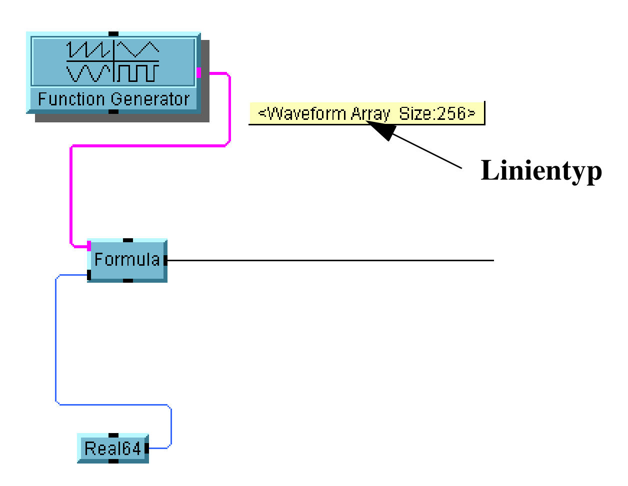


Abbildung 2-21. Anzeigen des Werts an einem Ausgangs-Pin

Wenn Sie eine Datenlinie anklicken, erscheint ein Dialogfenster mit allen Informationen zu den Daten an der Linie. Abbildung 2-22 zeigt beispielsweise das Dialogfenster an, das erscheint, wenn Sie den Ausgang des `Function Generator` anklicken.

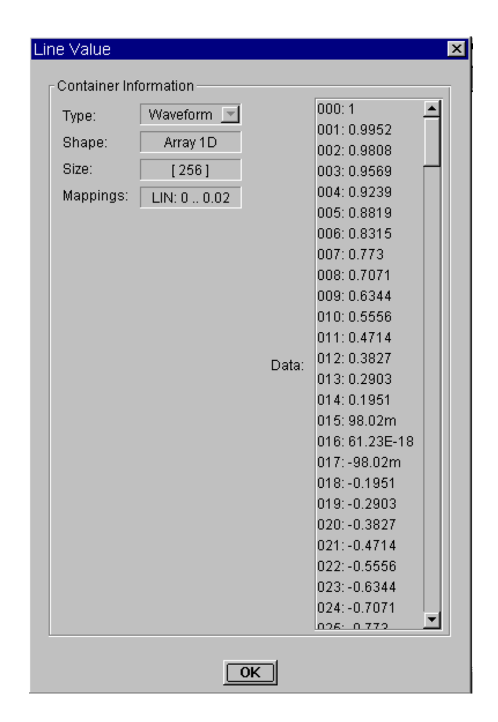


Abbildung 2-22. Anzeigen von Informationen zu einer Linie

Überprüfen von Anschlüssen

Zum Überprüfen eines Anschlusses klicken Sie diesen doppelt in der offenen Ansicht an, wie im Abschnitt “Informationen zu Pins und Anschlüssen” auf Seite 49 beschrieben. Wenn ein Objekt als Symbol angezeigt wird, positionieren Sie den Mauszeiger auf den Anschluss. VEE blendet daraufhin automatisch den Namen des Anschlusses ein.

Verwenden der alphanumerischen Anzeigen für die Fehlerbehebung

Sie können die Anzeigen `Alphanumeric` oder `Logging Alphanumeric` an verschiedenen Punkten in in einem Programm hinzufügen, um den Datenfluss zu verfolgen. Wenn das Programm richtig ausgeführt wird, löschen Sie diese Anzeigen wieder. `AlphaNumeric` zeigt einen einzelnen

Daten-Container (einen `Scalar`-Wert, einen `Array 1D` oder einen `Array 2D`) an, und `Logging AlphaNumeric` (ein `Scalar` oder ein `Array 1D`) zeigt die fortlaufende Eingabe als Historie früherer Werte an. Sie können auch einen `Counter` (Zähler) verwenden, um anzuzeigen, wie oft ein Objekt ausgeführt wurde.

Verwenden von Unterbrechungspunkten

Ein **Unterbrechungspunkt** bewirkt, dass ein Programm vor der Ausführung eines bestimmten Objekts angehalten wird. Sie können Unterbrechungspunkte in einem Programm setzen, um die Daten zu überprüfen. Wenn ein Unterbrechungspunkt auf einem Objekt gesetzt wird, wird das Objekt mit einer orangefarbenen Umrisslinie hervorgehoben. Bei der Ausführung des Programms wird es vor der Ausführung dieses Objekts angehalten.

1. Setzen Sie einen Unterbrechungspunkt auf einem einzelnen Objekt. Klicken Sie doppelt auf die Titelleiste eines Objekts, um das Dialogfenster `Properties` aufzurufen, wählen Sie `Breakpoint Enabled` (Unterbrechungspunkt aktiviert) aus und klicken Sie `OK` an. Wählen Sie anschließend `Debug` ⇒ `Activate Breakpoints` aus. Führen Sie das Programm aus. Es wird an dem gesetzten Unterbrechungspunkt angehalten.

2. Setzen Sie weitere Unterbrechungspunkte an verschiedenen weiteren Objekten. Wählen Sie die Objekte aus. (Drücken Sie **Strg** und klicken Sie die Objekte nacheinander an.) Klicken Sie die Schaltfläche **Unterbrechungspunkte umschalten** in der Symbolleiste an (siehe Abbildung 2-23). (Sie können stattdessen auch **Strg-B** drücken.) Führen Sie das Programm erneut aus. Das Programm wird an dem ersten Objekt mit einem gesetzten Unterbrechungspunkt angehalten.

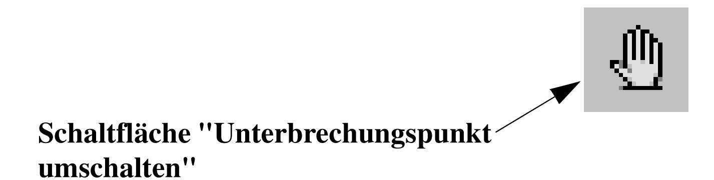


Abbildung 2-23. Unterbrechungspunkte setzen

3. Setzen Sie das Programm fort, sodass es am nächsten Objekt mit einem Unterbrechungspunkt angehalten wird. Klicken Sie die Schaltfläche **Fortsetzen** in der Symbolleiste an (siehe Abbildung 2-24). (Oder verwenden Sie die entsprechende Funktion im Menü `Debug`.)



Abbildung 2-24. Programm fortsetzen (gleiche Position wie Schaltfläche "Ausführen")

4. Löschen Sie jetzt die Unterbrechungspunkte aus dem Programm. Wählen Sie die Objekte mit Unterbrechungspunkten aus. Klicken Sie die Schaltfläche **Unterbrechungspunkte umschalten** in der Symbolleiste an (siehe Abbildung 2-25). Sie können stattdessen auch `Debug` ⇒ `Clear All Breakpoints` auswählen.



Abbildung 2-25. Unterbrechungspunkt(e) löschen

5. Zum Anhalten oder Stoppen des Programms klicken Sie die Schaltfläche **Pause** bzw. **Stopp** in der Symbolleiste an (siehe Abbildung 2-26). (Sie können auch die entsprechende Funktion im Menü `Debug` verwenden.)

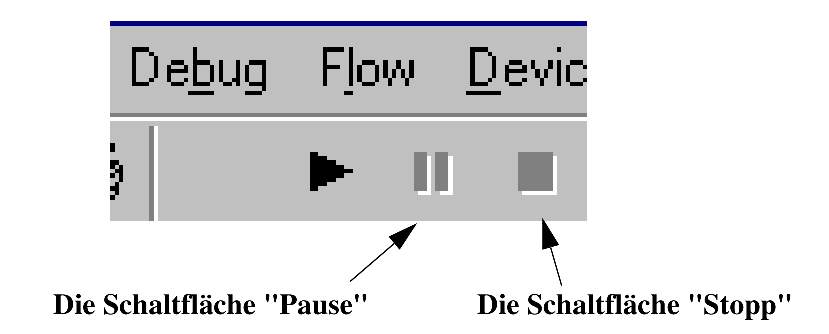


Abbildung 2-26. Programm anhalten oder stoppen

Beheben von Fehlern

Wenn bei der Ausführung eines Programms eine Fehlermeldung angezeigt wird, zieht VEE automatisch eine rote Umrisslinie um das Objekt, bei dem der Fehler festgestellt wurde.

Sie können entweder den Fehler korrigieren, damit die Umrisslinie verschwindet, oder Sie können die Schaltfläche **Stopp** anklicken; in diesem Fall wird die rote Umrisslinie ausgeblendet, und Sie können jetzt den Fehler beheben. Wenn Sie **Stopp** anklicken, können Sie den Fehler vor der Fortsetzung mit `View ⇒ Last Error` noch einmal anzeigen.

Verwenden der Schaltfläche "Go To" zum Suchen eines Fehlers

Abbildung 2-27 zeigt ein Beispiel für eine Laufzeit-Fehlermeldung. Bei der Ausführung dieses Programms zeigt VEE einen Laufzeitfehler an und zieht eine rote Umrisslinie um das `UserObject AddNoise`. Wenn Sie die Schaltfläche `Go To` anklicken, öffnet VEE das `UserObject AddNoise` und zeigt eine rote Umrisslinie um das Objekt `A + B` an; an diesem Objekt fehlt eine Verbindung am Eingangs-Pin `A`. In einem großen Programm kann die Funktion `Go To` die schnelle Suche nach einem Fehler erheblich erleichtern.

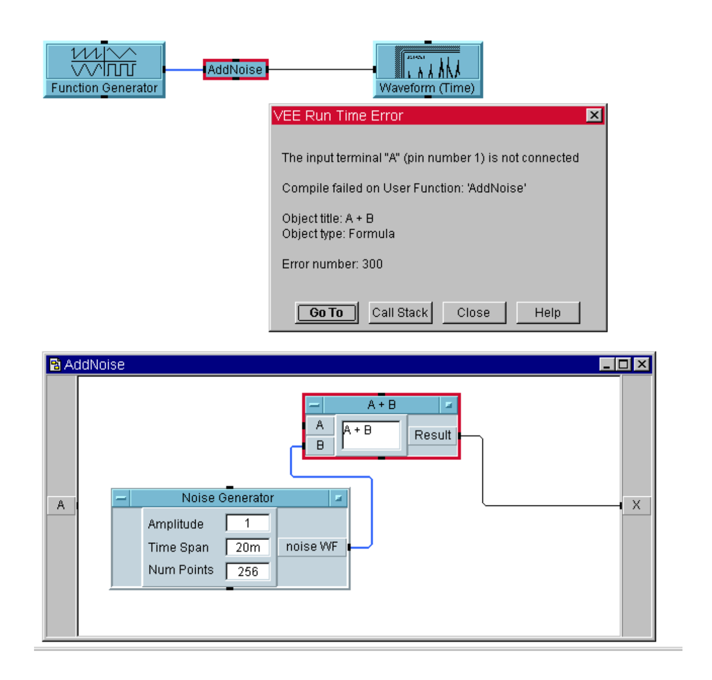


Abbildung 2-27. Beispiel einer Laufzeitfehlermeldung mit Go To

Verwenden des Aufrufstapels

Wenn ein Fehler im Hauptprogramm vorliegt, ist er meist recht leicht zu sehen. In einem großen Programm erleichtert der `Aufrufstapel` ("Call Stack) das Finden von Fehlern, die in mehreren Ebenen verschachtelt sind.

1. Klicken Sie die Schaltfläche **Pause** in der Symbolleiste (neben der Schaltfläche **Ausführen**) an.

2. Drücken Sie die Schaltfläche `Call Stack` im Fehler-Dialogfenster oder wählen Sie `View` ⇒ `Call Stack` aus. `Call Stack` listet die Hierarchie der Programmausführung auf.

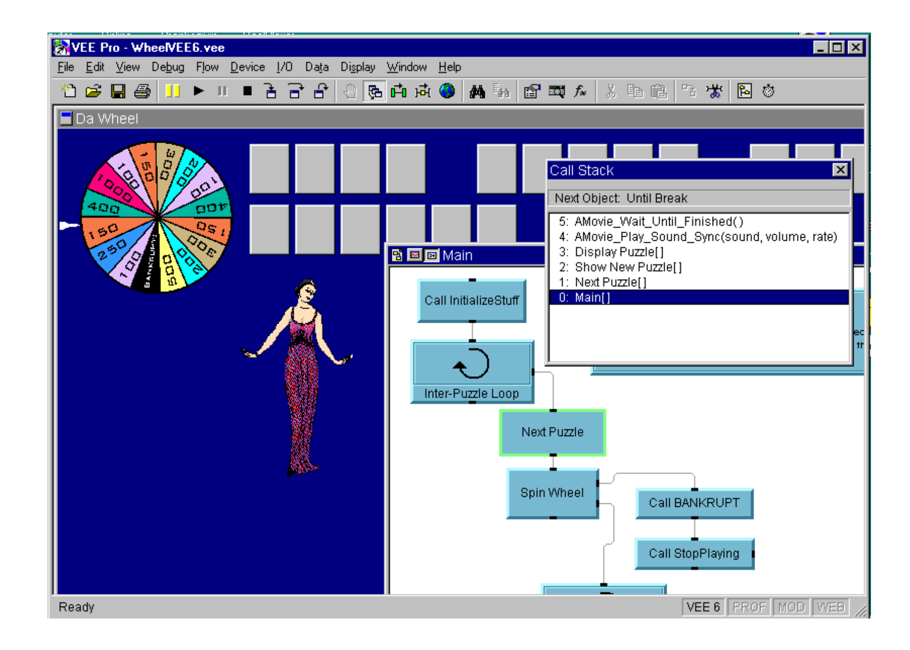


Abbildung 2-28. Verwendung des Aufrufstapels in Wheel.exe

Der `Aufrufstapel` zeigt die Hierarchie der Programmausführung an. Abbildung 2-28 zeigt ein Beispielprogramm, das zusammen mit VEE ausgeliefert wird: das Programm `Wheel.exe` in `Examples/Games`. In Abbildung 2-28 führt das Programm gerade die Benutzerfunktion `AMovie_Wait_Until_Finished()` aus, die von `AMovie_Play_Sound_Sync` aufgerufen wurde, die wiederum von `...Next_Puzzle` im Hauptprogramm aufgerufen wurde. Sie können auf jedes beliebige Element im Aufrufstapel doppelt klicken, damit VEE die entsprechende Funktion sucht und anzeigt.

Verfolgen der Ereignisreihenfolge in einem Objekt

Abbildung 2-29 zeigt die Reihenfolge von Ereignissen in einem Objekt.

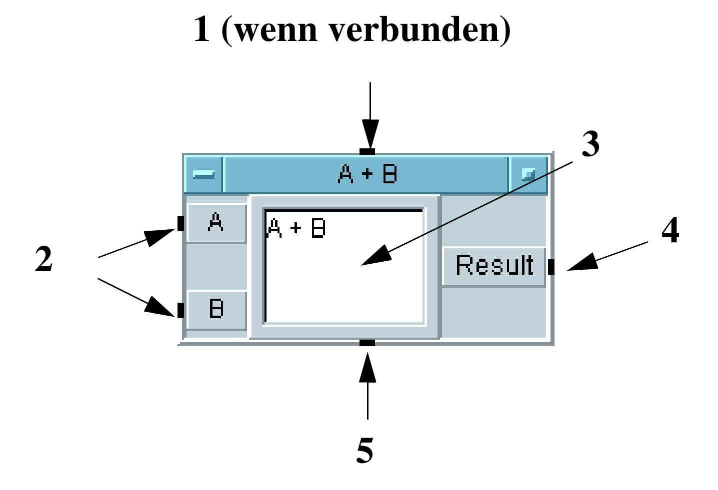


Abbildung 2-29. Die Reihenfolge der Ereignisse in einem Objekt

In Abbildung 2-30 arbeiten die Pins wie folgt:

1. Wenn der Sequenz-Pin verbunden ist, wird das Objekt nicht ausgeführt, bevor es eine Meldung erhält, die es zur Ausführung auffordert (ein “ping”, um die VEE-Terminologie zu verwenden). Der Sequenzeingangs-Pin *braucht jedoch nicht* angeschlossen zu sein.

2. Alle Dateneingangs-Pins müssen Daten enthalten, bevor das Objekt ausgeführt werden kann. (Sie können den meisten Objekten Datenein-/ausgangs-Pins hinzufügen. Klicken Sie das Menü `Anschluss hinzufügen/löschen` in einem beliebigen Objektmenü an, um festzustellen, welche Pins hinzugefügt werden können.)

3. Das Objekt führt seine Aufgabe aus. In diesem Fall wird `A` dem Objekt `B` hinzugefügt, und das Ergebnis wird am Ausgangs-Pin angelegt.

4. Der Datenausgangs-Pin löst aus. Das Objekt wartet auf ein Signal vom nächsten Objekt, dass die Daten empfangen wurden, bevor die Operation abgeschlossen wird. Ein gegebenes Objekt löst daher seinen Sequenzausgangs-Pin erst aus, wenn alle an seinem Datenausgangs-Pin angeschlossenen Objekte Daten empfangen haben.
5. Der Sequenzausgangs-Pin löst aus.

Es gibt einige Ausnahmen zu dieser Reihenfolge von Ereignissen:

- Sie können Fehler-Ausgangs-Pins hinzufügen, um Fehler innerhalb eines Objekts zu erfassen. Die Fehler-Ausgangs-Pins überschreiben das Verhalten des Standardobjekts. Wenn bei der Ausführung des Objekts ein Fehler auftritt, sendet der Fehler-Pin eine Meldung, und die Datenausgabe-Pins werden nicht ausgelöst.

- Sie können manchen Objekten Steuereingangs-Pins hinzufügen; diese können bewirken, dass das Objekt eine sofortige Aktion ausführt. Eine Unterfunktion eines Objekts wie `Title` oder `Autoscale` in der Anzeige `Waveform (Time)` kann beispielsweise mit Steuer-Pins ausgeführt werden. Steuerlinien zu einem Objekt werden in VEE-Programmen als gestrichelte Linien angezeigt.

  Abbildung 2-30 zeigt beispielsweise eine Steuerlinie, die einen angepassten Titel für die Wellenformanzeige festlegt. Beachten Sie, dass das Objekt zur Ausführung dieser Aktion keine Daten an einem Steuer-Pin benötigt. Das Objekt wird nicht ausgeführt; die einzige ausgeführte Aktion ist das Festlegen des Titels. Sie können `Show Data Flow` anklicken, um zu sehen, wie die Steuerlinie zum Steuereingang `Title` zuerst Daten überträgt.

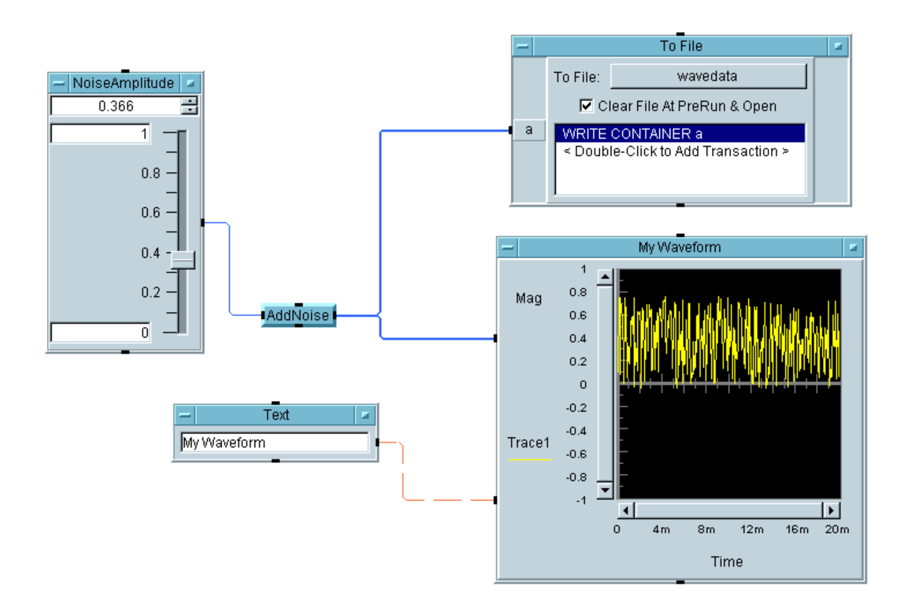


Abbildung 2-30. Steuerlinie zur Ausführung eines angepassten Titels

Verfolgen der Ausführungsreihenfolge von Objekten in einem Programm

Bei der Ausführung eines VEE-Programms werden die Objekte in der folgenden Reihenfolge ausgeführt:

1. `Start`objekte werden zuerst ausgeführt.

   Abbildung 2-32 zeigt ein VEE-Programm mit zwei **Threads**; hierbei handelt es sich um Gruppen von Objekten, die mit durchgezogenen Linien in einem VEE-Programm verbunden sind. Die `Start`objekte unter `Flow` ⇒ `Start` werden zur Ausführung der einzelnen Threads in einem Programm verwendet. Wenn ein Programm `Start`objekte enthält, werden diese zuerst ausgeführt.

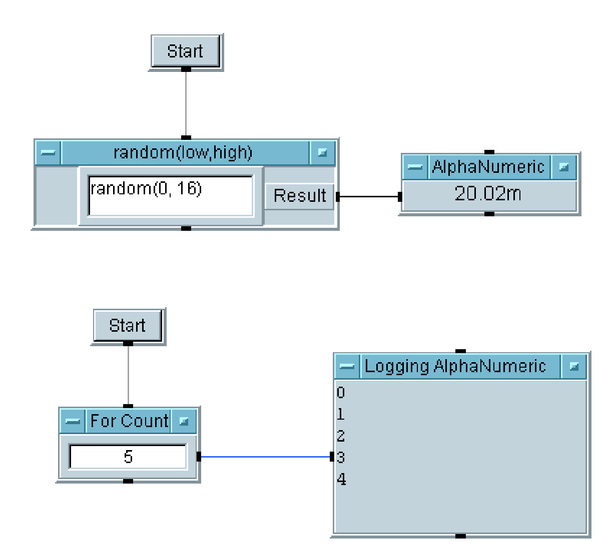


Abbildung 2-31. Startobjekte zur Ausführung separater Threads

2. Objekte ohne Dateneingangs-Pins werden als nächste ausgeführt. `Data ⇒ Constant`-Objekte gehören häufig zu dieser Kategorie.

3. Objekte mit Eingangs-Pins werden nur ausgeführt, wenn an allen angeschlossenen Eingängen Daten anliegen. (Denken Sie daran, dass das Anschließen von Sequenzeingängen optional ist.)

Schrittweise Ausführung eines Programms

Die schrittweise Ausführung eines Programms ist eine sehr effektive Methode zur Fehlerbehebung. VEE enthält die Funktionen `Schritt in`, `Schritt über` und `Schritt aus`, mit denen Sie in, über und aus einem Objekt springen können.

Zum Aktivieren der Schrittfunktion klicken Sie eine der Schaltflächen **Schritt in**, **Schritt über** oder **Schritt aus** in der Symbolleiste an (siehe Abbildung 2-32).

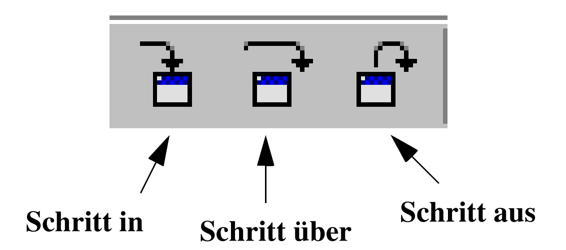


Abbildung 2-32. Schaltflächen "Schritt in", "Schritt über" und "Schritt aus" in der Symbolleiste

- **Schritt in** führt ein Programm ein Objekt nach dem anderen aus. Wenn das Programm ein `UserObject` oder eine `UserFunction` erreicht, versetzt VEE das `UserObject` bzw. die `UserFunction` in die Detailansicht und führt die darin enthaltenen Objekte aus.

- **Schritt über** und **Schritt aus** führen ein Programm ein Objekt nach dem anderen aus, ohne `UserObjects` oder `UserFunctions` zu öffnen. Wenn das Programm ein `UserObject` oder eine `UserFunction` erreicht, führt VEE das `UserObject` bzw. die `UserFunction` vollständig aus.

Beispiel zur schrittweisen Ausführung eines Programms:

1. Öffnen Sie das Programm `simple-program_with_panel.vee`.

2. Klicken Sie die Schaltfläche **Schritt in** in der Symbolleiste an.

3. Beim wiederholten Anklicken von **Schritt in** durchlaufen die farbigen Umrisslinien um die Objekte das Programm sequenziell.

Bei der schrittweisen Ausführung legt VEE die `Fensteransicht` hinter der `Detailansicht` ab, um die Reihenfolge der Objekte bei der Ausführung anzeigen zu können. Innerhalb des `Hauptprogramms` sind an den Eingangsfeldern keine Eingangs-Pins angeschlossen, sie werden daher ohne bestimmte Reihenfolge ausgeführt. Wenn diese Felder in einer bestimmten Reihenfolge ausgeführt werden sollen, können Sie hierzu ihre Sequenz-Pins anschließen.

Der Datenfluss erfolgt von links nach rechts, die Datengeneratoren werden daher als nächstes ohne eine bestimmte Reihenfolge ausgeführt. Das Additionsobjekt (`A+B`) kann nicht ausgeführt werden, bis beide Eingänge Daten empfangen haben. Anschließend wird das Objekt `Waveform Time)` ausgeführt. Auch hierbei können Sie die Ausführung an einer bestimmten Stelle im Programm über die Sequenz-Pins oder das Objekt `Flow` ⇒ `Do` festlegen. (Weitere Informationen über das Objekt `Do` finden Sie in der Online-Hilfe.)

Hinweis

Weitere Informationen zu den Schrittfunktionen finden Sie in der Online-Hilfe. Weitere Informationen zu `UserFunctions` finden Sie in Kapitel 8, “Verwenden von Agilent VEE-Funktionen” auf Seite 309.

Suchen eines Objekts in einem komplexen Programm

Zum Suchen eines bestimmten Objekts, insbesondere in einem großen Programm, wählen Sie `Edit` ⇒ `Find` aus. Geben Sie den Objekt- oder Funktionsnamen im Einblend-Dialogfenster ein, und VEE zeigt alle Exemplare und Positionen dieses Objekts bzw. dieser Funktion in dem Programm an. (Weitere Informationen siehe “Finden von Funktionen in großen Programmen” auf Seite 337.)

Übungen mit Programmen

Die Übungsprogramme in diesem Abschnitt verdeutlichen weitere VEE-Funktionen.

Übung 2-6: Generieren einer Zufallszahl

1. Dokumentieren Sie das Programm.

   a. Wählen Sie `Display ⇒ Note Pad` aus und platzieren Sie diesen Notizblock oben in der Mitte des Arbeitsbereichs. Klicken Sie den Bearbeitungsbereich an, um einen Cursor zu erhalten, und geben Sie ein:

   ```
   This program, Random, generates a real number
   between 0 and 1, then displays the results.
   ```

2. Wählen Sie `Device ⇒ Function & Object Browser` aus. Wählen Sie `Type ⇒ Built-in` function, `Category ⇒ All` und `Functions ⇒ random` aus. Klicken Sie `Create Formula` an. Positionieren Sie das Objekt im Arbeitsbereich und klicken Sie mit der Maustaste, um das Objekt zu platzieren.

3. Klicken Sie `Data ⇒ Constant ⇒ Int32` an und platzieren Sie das Objekt links von `random`. Öffnen Sie das `Int32`-Objektmenü, klicken Sie `Clone` an und legen Sie dieses Objekt unter dem anderen `Int32` ab. Klicken Sie doppelt auf `0`, um einen Cursor anzuzeigen, und geben Sie `1` ein. Verbinden Sie das Konstantenobjekt mit `0` am unteren Eingangs-Pin von `random`, und verbinden Sie die Konstante `1` mit dem oberen Eingangs-Pin.

4. Wählen Sie `Display ⇒ AlphaNumeric` aus und platzieren Sie das Objekt rechts von dem Objekt `random`. Öffnen Sie die Objektmenüs, und wählen Sie `Help` aus, um mehr über die Objekte zu erfahren.

5. Verbinden Sie den Ausgangs-Pin des Objekts `random` mit dem Eingangs-Pin von `AlphaNumeric`. Es wird eine Datenlinie angezeigt, die die beiden Objekte verbindet.

Hinweis

Beim Verschieben des Mauszeigers, während die Linie in der Nähe des Ziel-Pins angeschlossen ist, wird der Pin durch einen Rahmen hervorgehoben. Klicken Sie anschließend erneut, um die Verbindung abzuschließen.

Hinweis

Wenn Sie aus irgendeinem Grund die Operation zum Verbinden der Linie beenden wollen, bevor die Verbindung abgeschlossen wurde, klicken Sie doppelt mit der Maustaste - die Linie verschwindet.

6. Klicken Sie die Schaltfläche **Ausführen** in der Symbolleiste an, um eine Zufallszahl anzuzeigen (siehe Abbildung 2-33).

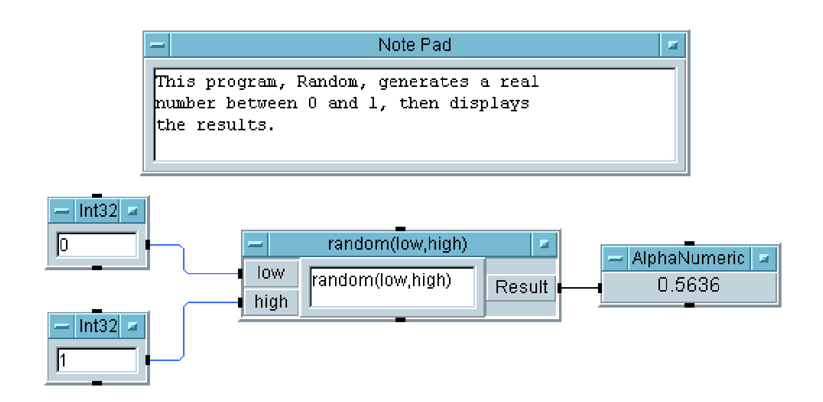


Abbildung 2-33. Das Programm "Random"

7. Wählen Sie `File` ⇒ `Save As...` aus, geben Sie `Random.VEE` ein und klicken Sie `OK` an. (Oder speichern Sie das Programm unter `EVAL.VEE`, falls Sie die Evaluation-Kit-Software verwenden.) Dieser Name wird beim nächsten Öffnen neben VEE in der Titelleiste angezeigt.

Übung 2-7: Einstellen und Abrufen einer globalen Variablen

Dieses Programm bietet Ihnen zusätzliche Übung mit den Grundfunktionen beim Erstellen eines VEE-Programms und erläutert das Prinzip der globalen Variablen. Sie können das Objekt `Set Variable` zum Erstellen einer

Variablen verwenden, die Sie später in dem Programm mit einem Objekt `Get Variable` abrufen können. Sie können einen beliebigen VEE-Datentyp verwenden. In diesem Beispiel wird eine Zahl des Typs `Real64` verwendet. (Weitere Informationen zu VEE-Datentypen finden Sie in Kapitel 4, “Analysieren und Anzeigen von Testdaten”)

1. Wählen Sie `Display ⇒ Note Pad` aus und platzieren Sie diesen Notizblock oben in der Mitte des Arbeitsbereichs. Klicken Sie in die linke obere Ecke des Editor-Bereichs, um einen Cursor anzuzeigen, und geben Sie die folgenden Informationen ein:

   ```
   Set and Get a Global Variable prompts the user to
   enter a real number. The variable, num, is set to this
   real number. Then num is recalled and displayed.
   ```

2. Wählen Sie `Data ⇒ Constant ⇒ Real64` aus und platzieren Sie das Objekt auf der linken Seite des Arbeitsbereichs. Öffnen Sie das Objektmenü und überprüfen Sie den `Help`- Eintrag.

3. Öffnen Sie das `Real64`-Objektmenü und wählen Sie `Properties` aus. Ändern Sie den Titel auf die Eingabeaufforderung `Enter a Real Number:` und klicken Sie `OK` an.

Hinweis

In dieser Übung wird eines der Konstantenobjekte für ein Eingangs-Dialogfenster verwendet durch die einfache Änderung des Titels in eine Eingabeaufforderung. Dies ist eine gebräuchliche Technik zum Abrufen von Benutzereingaben. Sie können `Data ⇒ Dialog Box ⇒ Real64 Input` verwenden. Außerdem können Sie auch die Titelleiste doppelt anklicken, um das Dialogfenster `Constant Properties` abzurufen.

4. Wählen Sie `Data ⇒ Variable ⇒ Set Variable` aus und platzieren Sie das Objekt rechts von dem Objekt `Real64`. Klicken Sie doppelt auf `globalA`, um es hervorzuheben, und geben Sie anschließend `num` ein. Beachten Sie, dass sich der Name des Objekts in `Set num` ändert.

   Dies bedeutet, dass der Benutzer eine reale Zahl im Objekt `Real64` eingibt. Wenn der Benutzer die Schaltfläche **Ausführen** anklickt, wird die Zahl auf die globale Variable `num` gesetzt.

5. Verbinden Sie den Datenausgangs-Pin des Objekts `Real` mit dem Dateneingangs-Pin des Objekts `Set num`.

6. Wählen Sie `Data ⇒ Variable ⇒ Get Variable` aus und platzieren Sie das Objekt unter dem Objekt `Set num`. Ändern Sie den Variablennamen in `num`. Beachten Sie, dass sich der Name des Objekts in `Get num` ändert.

7. Verbinden Sie den Sequenzausgangs-Pin von `Set num` mit dem Sequenzeingangs-Pin von `Get num`.

Hinweis Eine globale Variable muss gesetzt werden, bevor Sie sie verwenden können. Sie müssen daher in diesem Fall die Sequenz-Pins verwenden, um sicherzustellen, dass die Variable "num" gesetzt wurde, bevor Sie sie mit `Get num` abrufen.

8. Wählen Sie `Display ⇒ AlphaNumeric` aus und platzieren Sie das Objekt rechts von dem Objekt `Get num`.

9. Verbinden Sie den Datenausgangs-Pin des Objekts `Get num` mit dem Dateneingangs-Pin von `AlphaNumeric`.

10. Geben Sie eine reale Zahl ein und klicken Sie die Schaltfläche "Ausführen" in der Symbolleiste an. Das Programm sollte jetzt aussehen wie in Abbildung 2-34.

11. Wählen Sie `File ⇒ Save As...` aus, und nennen Sie das Programm `global.vee`. (Wenn Sie die Evaluation-Kit-Software verwenden, speichern Sie das Programm unter dem Namen `EVAL.VEE`.)

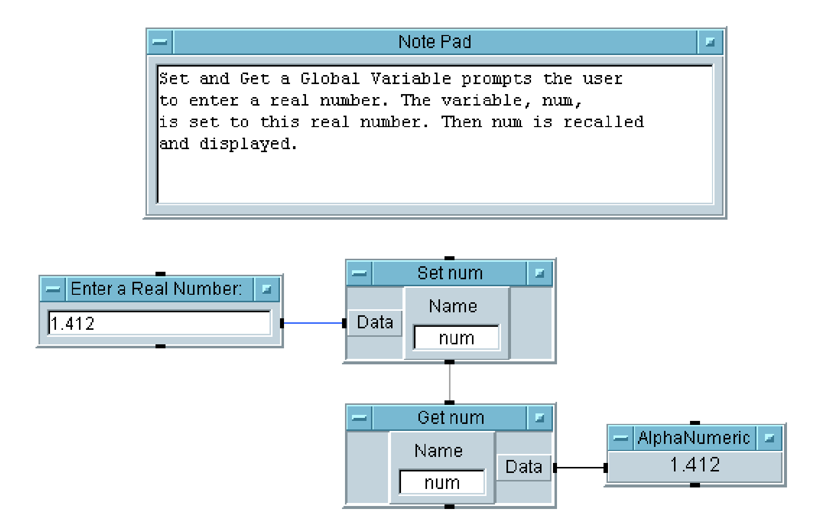


Abbildung 2-34. Setzen und Abrufen einer globalen Variablen

Dokumentieren von Agilent VEE-Programmen

Durch die Verwendung des Befehls `File ⇒ Save Documentation...` können Sie automatisch die Dokumentation zu Ihrem Programm erstellen. VEE listet alle Objekte mit ihren wichtigsten Einstellungen, den Standard- und Benutzernamen, den Beschreibungseinträgen und allen Verschachtelungen auf. Objekt innerhalb eines `UserObject` sind beispielsweise eine Ebene von der VEE-Hauptumgebung aus verschachtelt; diese Ebenen werden mit Nummern gekennzeichnet.

Sie können auch einzelne Objekte mit einer Beschreibung dokumentieren. Zunächst wird in dieser Übung erläutert, wie ein einzelnes Objekt dokumentiert wird; anschließend wird das Generieren der Programmdokumentation beschrieben.

Dokumentieren von Objekten mit Beschreibungs-Dialogfenstern

Alle Objekte enthalten ein Element `Description` (Beschreibung) in ihren Objektmenü, das ein Dialogfenster bereitstellt, über das eine Dokumentation zu dem entsprechenden Objekt eingegeben werden kann. Diese Dokumentationsdatei bietet außerdem die Möglichkeit, die Dokumentation mit den Bildschirmauszügen abzustimmen. In diesem Abschnitt fügen Sie dem Dialogfenster `Description` einen Eintrag hinzu.

1. Öffnen Sie das Programm `Random.vee`.

2. Klicken Sie im Haupt-Objektmenü `Description` an. Geben Sie den Text in dem Dialogfeld ein (siehe Abbildung 2-36): Sobald die Angaben vollständig sind, klicken Sie `OK` an.

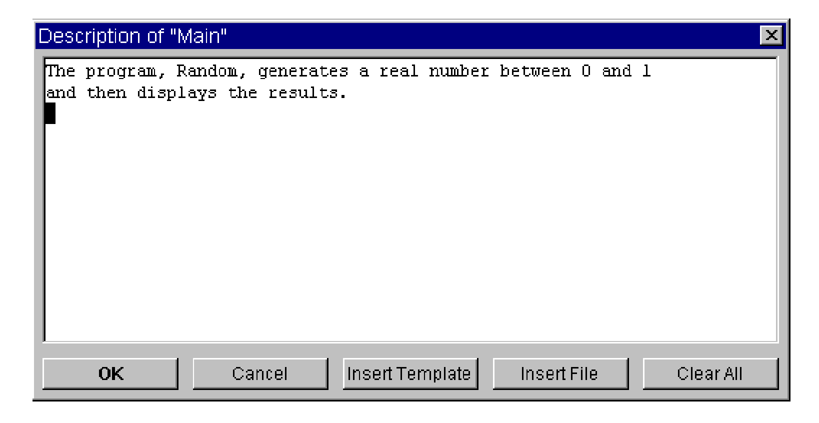


Abbildung 2-35. Das Dialogfenster "Description"

Hinweis

Die Einträge in dem Dialogfenster `Description` sind für die Benutzer nicht sichtbar, es sei denn, sie rufen sie über das Objektmenü auf. Beachten Sie auch, dass Sie in diesem Dialogfenster eine Datei oder eine Vorlage einfügen können.

Automatisches Generieren der Dokumentation

Gehen Sie anhand der folgenden Schritte vor, um eine Datei mit der Programmdokumentation zu generieren:

1. Öffnen Sie das Programm `Random.vee`. Klicken Sie `File` ⇒ `Save Documentation...` an. Geben Sie den Dateinamen mit einem Suffix `*.txt` (z. B. `Random.txt`) ein und klicken Sie `Save` an. Standardmäßig wird die Datei auf dem PC in dem Ordner `C:\My Documents\VEE Programs` gespeichert.

2. Öffnen Sie die Datei in einem beliebigen Texteditor zum Anzeigen oder Drucken. Abbildung 2-36, Abbildung 2-37 und Abbildung 2-38 zeigen die Dokumentationsdatei mit dem Notepad-Programm in MS Windows 98 an.

   Abbildung 2-36 zeigt den Anfang der Datei mit Informationen zu der Datei, dem Datum der Änderungen und der System-E/A-Konfiguration.

```
Source file: "C:\\My Documents\\VEE Programs\\Random.vee"
File last revised: Mon Jan 03 15:29:02 2000
Date documented:  Mon Feb 28 14:43:27 2000
VEE revision: 6.0
Execution mode: VEE 6
Convert Infinity on Binary Read: no

I/O Configuration
  My Configuration (C:\WINDOWS\Local Settings\Application
Data\Agilent VEE\vee.io)
```

Abbildung 2-36. Der Anfang der Dokumentationsdatei

```
M: Main
 Device Type              : Main
 Description              :
   1. The program, Random, generates a real numer between 0 and 1
   2. and then displays the results.
 Context is secured       : off
 Trig mode                : Degrees
 Popup Panel Title Text   : Untitled
 Show Popup Panel Title   : on
 Show Popup Panel Border  : on
 Popup Moveable           : on
 Popup Panel Title Text Color         : Object Title Text
 Popup Panel Title Background Color   : Object Title
 Popup Panel Title Text Font          : Object Title Text
 Delete Globals at Prerun : on

M.0: Main/Note Pad
 Device Type              : Note Pad
 Note Contents            :
   1. This program, Random, generates a real
   2. number between 0 and 1, then displays
   3. the results.
   4.

M.1: Main/random(low,high)
 Device Type              : Formula
  Input pin  1            : low (Any, Any)
  Input pin  2            : high (Any, Any)
  Output pin 1            : Result
 Formula                  : random(low,high)
```

Abbildung 2-37. Die Mitte der Dokumentationsdatei

In Abbildung 2-37 sind die VEE-Objekte zusammen mit ihren Einstellungen beschrieben. Die Nummer vor den einzelnen Objekten gibt an, wo sich das Objekt befindet. Das erste Objekt im Hauptprogramm ist beispielsweise als `M1` aufgelistet. Abbildung 2-38 zeigt den Rest dieser Dokumentationsdatei als Referenz.

```
M.2: Main/Int32
 Device Type                : Constant
  Output pin 1              : Int32
 Wait For Event             : off
 Auto execute               : off
 Initialize At Prerun       : off
 Initialize at Activate     : off
 Constant size fixed        : off
 Password masking           : off
 Indices Enabled            : on
 Int32 Value                : 0
M.4: Main/Int32
 Device Type                : Constant
  Output pin 1              : Int32
 Wait For Event             : off
 Auto execute               : off
 Initialize At Prerun       : off
 Initialize at Activate     : off
 Constant size fixed        : off
 Password masking           : off
 Indices Enabled            : on
 Int32 Value                : 1
M.5: Main/AlphaNumeric
 Device Type                : AlphaNumeric
  Input pin  1              : Data (Any, Any)
 Clear At Prerun            : on
 Clear at Activate          : on
 Indices Enabled            : on
```

Abbildung 2-38. Der Rest der Dokumentationsdatei

Hinweis Führen Sie nach der Ausführung des Befehls `Save Documentation` einen Befehl `File ⇒ Print Program` aus, um die Objekte mit Identifikationsnummern zu versehen, damit Sie die Textdokumentation mit der Druckausgabe abstimmen können.

Kapitel-Checkliste

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die Themen ggf. noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Erstellen eines `UserObject` und Erläutern, wie `UserObjects` Programmen eine Struktur verleihen und Platz auf dem Bildschirm sparen.
- Erstellen von Einblend-Dialogfenstern und Reglern (oder Knöpfen) für die Benutzereingabe.
- Verwenden von Datendateien zum Speichern von Daten in einer Datei und Laden von Daten aus einer Datei.
- Erstellen einer Bedienerschnittstelle mit einer Fensteransicht des Programms.
- Verwenden verschiedener Datentypen und Datenformen.
- Verwenden mathematischer Operatoren und Funktionen.
- Verwenden der `Online-Hilfe`.
- Anzeigen des Datenflusses und des Ausführungsflusses in einem Programm.
- Beheben von Fehlern in einem Programm durch Untersuchen von Daten an einer Linie, Anschlüssen und alphanumerischen Anzeigen.
- Verwenden von Unterbrechungspunkten.
- Auflösen von Fehlern mit dem Befehl `GoTo`.
- Auflösen von Fehlern mit dem Aufrufstapel.
- Verwenden von Schritt in, Schritt über und Schritt aus zum Verfolgen und Beheben von Fehlern in einem Programm.
- Verwenden der Suchfunktion.
- Dokumentieren von Objekten mit Beschreibungs-Dialogfenstern.
- Generieren einer Dokumentationsdatei.

3

Einfache Methoden zum Steuern von Instrumenten

Einfache Methoden zum Steuern von Instrumenten

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Konfigurieren von Instrumenten
- Verwenden eines Panel-Treibers
- Verwenden des Objekts "Direct I/O"
- Steuern von PC-Zusatzkarten
- Verwenden eines VXI*Plug&Play*-Treibers

Erforderliche Zeit für dieses Kapitel: 1 Stunde

Überblick

In diesem Kapitel lernen Sie, wie Sie mit VEE Instrumente steuern können. Mit VEE können Sie Instrumente auf verschiedene Arten steuern:

- **“Panel”-Treiber** bieten eine einfache Benutzerschnittstelle (ein “Bedienfeld”) zum Steuern eines Instruments von Ihrem Computerbildschirm aus. Wenn Sie Parameter im VEE-Panel-Treiber ändern, wird der entsprechende Status des Instruments geändert. Panel-Treiber werden von Agilent Technologies zusammen mit VEE bereitgestellt und decken mehr als 450 Instrumente von verschiedenen Herstellern ab.

- **Das Objekt "Direct I/O"** ermöglicht das Übertragen von Befehlen und das Empfangen von Daten über eine Reihe unterstützter Schnittstellen. Diese Technik entspricht der Verwendung von Befehlsfolgen mit einer Textsprache wie beispielsweise Rocky Mountain Basic.

- **Open Data Acquisition Standard Drivers (ODAS-Treiber)** verwenden die ActiveX-Automatisierungs-Technologie zum Steuern von PC-Zusatzkarten (PCPI-Karten). Ein ODAS-Treiber hat ein Standardformat und kann daher vom Hersteller der PC-Zusatzkarte oder einem anderen Hersteller zur Verfügung gestellt werden.

- **I/O libraries** (E/A-Bibliotheken) können zum Steuern von PC-Zusatzkarten importiert werden, um über das Objekt `Call` Funktionen aus dieser Bibliothek aufzurufen. Diese Bibliotheken, meist als Dynamic Link Libraries (DLLs) ausgeliefert, ähneln ODAS-Treibern; allerdings sind ODAS-Treiber standardisiert und einfacher in der Handhabung.

- **VXI*Plug&Play*-Treiber** können zum Aufruf von C-Funktionen zum Steuern von Instrumenten verwendet werden. Diese Treiber werden von Agilent Technologies und anderen Herstellern mit ihren unterstützten Instrumenten bereitgestellt.

Dieses Kapitel vermittelt Ihnen die Grundkenntnisse zum Steuern von Instrumenten für die meisten Situationen. Ausführlichere Informationen finden Sie in *VEE Pro Advanced Techniques*.

Panel-Treiber

Agilent VEE enthält über 450 Panel-Treiber für Instrumente von verschiedenen Herstellern. Ein Panel-Treiber verwendet eine Anzeige im VEE-Programm, das die Einstellung in dem entsprechenden physischen Instrument steuert. Panel-Treiber bieten eine maximale Benutzerfreundlichkeit und sparen viel Zeit bei der Entwicklung. Abbildung 3-1 zeigt ein Beispiel eines Panel-Treibers.

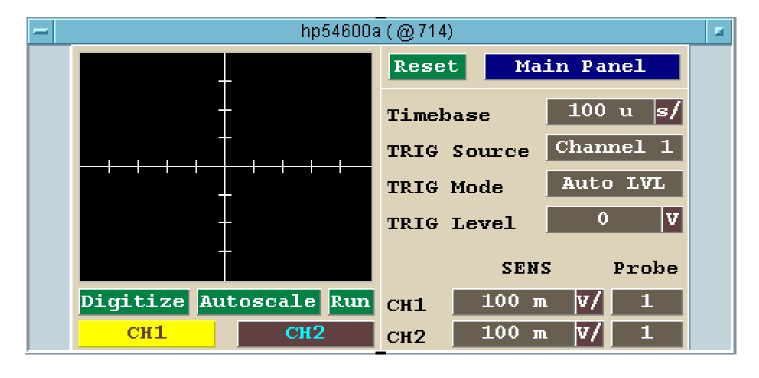


Abbildung 3-1. Der Panel-Treiber für das Oszilloskop HP54600A

Das Objekt "Direct I/O"

Das VEE-Objekt "Direct I/O" ermöglicht die Kommunikation mit einem beliebigen Instrument von einem beliebigen Hersteller über Standardschnittstellen (ob ein Treiber für das Instrument verfügbar ist oder nicht). Das Objekt "Direct I/O" arbeitet durch Übertragen von Befehlen an das Instrument und Empfangen von Daten von dem Instrument. Die Verwendung von "Direct I/O" ermöglicht im Allgemeinen eine schnellere Ausführung. Die Auswahl der besten Methode zur Steuerung von Instrumenten hängt von der Verfügbarkeit der Treiber, der Notwendigkeit für eine schnelle Testentwicklung und den Leistungsanforderungen ab. Abbildung 3-2 zeigt ein Beispiel zur Verwendung von "Direct I/O" zur Steuerung eines Funktionsgenerators.

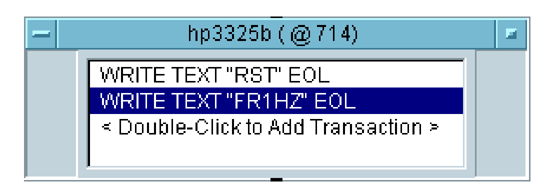


Abbildung 3-2. Ein Objekt "Direct I/O" für einen Funktionsgenerator

PC-Zusatzkarten mit ODAS-Treibern

ODAS-Treiber werden vom Hersteller der PC-Zusatzkarte bereitgestellt; da es sich hierbei um standardisierte Treiber handelt, können sie jedoch auch von anderen Herstellern stammen. VEE ermöglicht Ihnen die Steuerung einer PC-Zusatzkarte mit einem ODAS-Treiber durch Auswahl der PC-Zusatzkartenfunktionen in einem `Formula`-Objekt.

ODAS-Treiber bieten einen stärker standardisierten Weg zur Steuerung von PC-Zusatzkarten als herstellereigene DLLs; sie lassen sich auch besser von einem PC auf einen anderen portieren. Abbildung 3-3 zeigt ein Beispiel eines `Formula`-Objekts in VEE zur Steuerung einer PC-Zusatzkarte mit einem ODAS-Treiber.

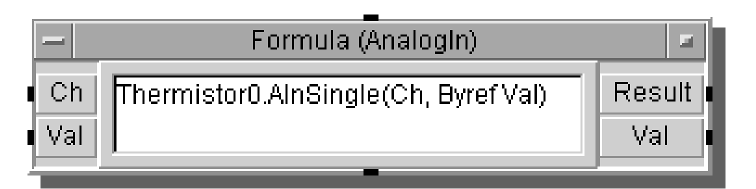


Abbildung 3-3. ODAS-Treiberobjekt in einem VEE-Programm

PC-Zusatzkarten mit E/A-Bibliothek

E/A-Bibliotheken, normalerweise in Form von Dynamically Linked Libraries (DLLs) für PC-Zusatzkarten ausgeliefert, werden vom Hersteller der PC-Zusatzkarte bereitgestellt. VEE ermöglicht die Steuerung von PC-Zusatzkarten durch den Aufruf von Bibliotheksfunktionen über das Objekt `Call`. Abbildung 3-4 zeigt ein Beispiel des Objekts `Import Library`, das die Funktionen in VEE verfügbar macht.

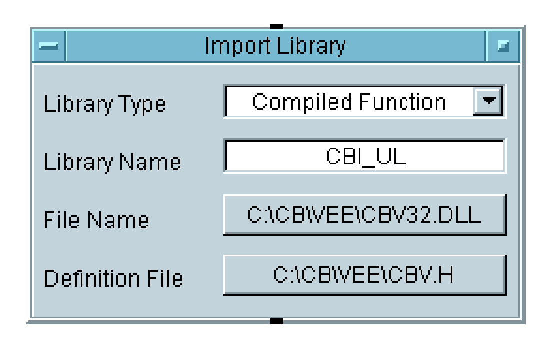


Abbildung 3-4. Importieren einer PC-Zusatzbibliothek

VXI*Plug&Play*-Treiber

VXI*Plug&Play*-Treiber werden vom Hersteller des Instruments bereitgestellt oder von Agilent Technologies. (Eine Liste der VXI*Plug&Play*-Treiber von Agilent Technologies finden Sie in der Literatur zu VEE oder im Handbuch *VEE Pro Advanced Techniques*. Wenden Sie sich bezüglich weiterer VXI*Plug&Play*-Treiber an den Hersteller Ihres Instruments.) VEE ermöglicht die Steuerung eines Instruments mit einem VXI*Plug&Play*-Treiber durch Aufrufe des Treibers. Abbildung 3-5 zeigt ein Beispiel für Aufrufe eines VXI*Plug&Play*-Treibers von VEE aus.

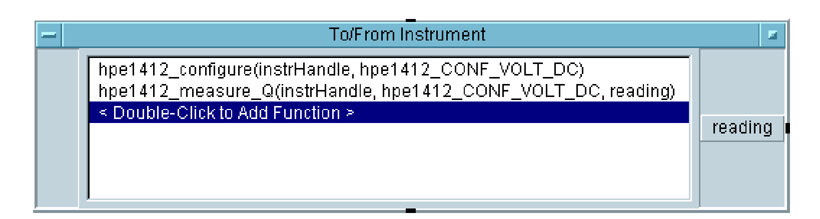


Abbildung 3-5. Aufrufe eines VXI*Plug&Play*-Treibers von VEE aus

Konfigurieren von Instrumenten

Mit VEE können Sie Programme entwickeln, ohne dass die Instrumente tatsächlich vorhanden sind. In dieser Übung konfigurieren Sie ein Oszilloskop zur Verwendung mit einem Panel-Treiber. Anschließend fügen Sie das physische Instrument der Konfiguration hinzu.

Übung 3-1: Konfigurieren eines Instruments, ohne dass das Instrument vorhanden ist

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus. Verschieben Sie das Dialogfenster in den Arbeitsbereich links oben, indem Sie seine Titelleiste anklicken und ziehen (siehe Abbildung 3-6).

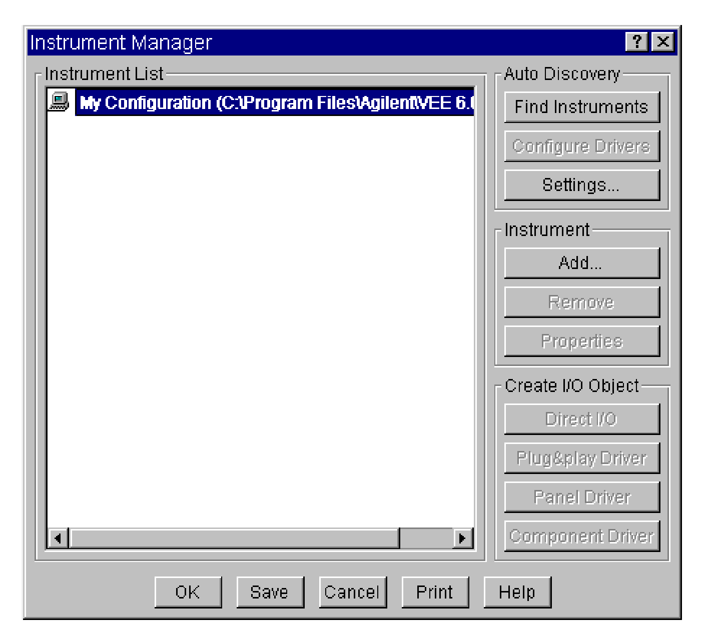


Abbildung 3-6. Das Fenster "Instrument Manager"

Hinweis

Wenn Sie Instrumente angeschlossen und eingeschaltet haben, kann VEE diese Instrumente und die passenden Treiber dazu automatisch finden. Weitere Informationen zum automatischen Suchen und Konfigurieren von Instrumenten finden Sie in den Online-Lernprogrammen unter `Help` ⇒ `Welcome` ⇒ `Tutorials` in der VEE-Hauptanzeige.

Standardmäßig sind keine Instrumente konfiguriert; in diesem Beispiel wird davon ausgegangen, dass keine Instrumente in der Liste des `Instrument Manager` angezeigt werden.

2. Vergewissern sie sich, dass im Dialogfenster `Instrument Manager` die Option `My Configuration` hervorgehoben ist, und klicken Sie `Add...` unter `Instrument` an. Das Dialogfenster `Instrument Properties` wird angezeigt (siehe Abbildung 3-7).

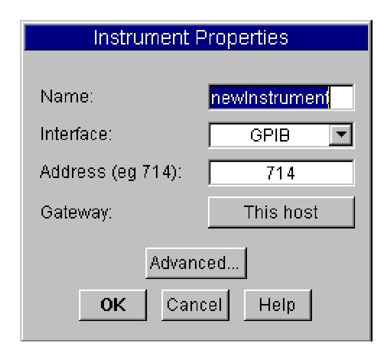


Abbildung 3-7. Dialogfenster "Instrument Properties"

Die Einträge im Dialogfenster `Instrument Properties` lauten wie folgt:

Name Der Name des Instruments wird im Programm aufgerufen. Wählen Sie einen Namen entsprechend den folgenden Syntaxvorgaben aus:

- Namen von Instrumenten müssen mit einem alphabetischen Zeichen beginnen, gefolgt von alphanumerischen Zeichen oder Unterstreichungszeichen.
- Sie können keine integrierten Leerzeichen in Instrumentnamen verwenden.

Interface Typ der Schnittstelle. Wählen Sie GPIB, Serial, GPIO oder VXI aus.

Address Die logische Einheit der Schnittstelle (GPIB verwendet normalerweise 7) plus der lokalen Busadresse des Instruments (eine Zahl von 0 bis 31). Wenn Sie die Adresse 0 beibehalten, bedeutet dies, dass bei der Entwicklung kein Instrument vorhanden ist.

Gateway Gibt an, ob Instrumente lokal oder fern gesteuert werden. Verwenden Sie den Standardeintrag `This host` zur lokalen Steuerung von Instrumenten, oder geben Sie einen `Gateway` für die ferne Steuerung ein. (Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

3. Ändern Sie den Namen in `scope`, behalten Sie alle weiteren Standardwerte bei, und klicken Sie `Advanced...` an. Das Dialogfenster `Advanced Instrument Properties` wird angezeigt (siehe Abbildung 3-8).

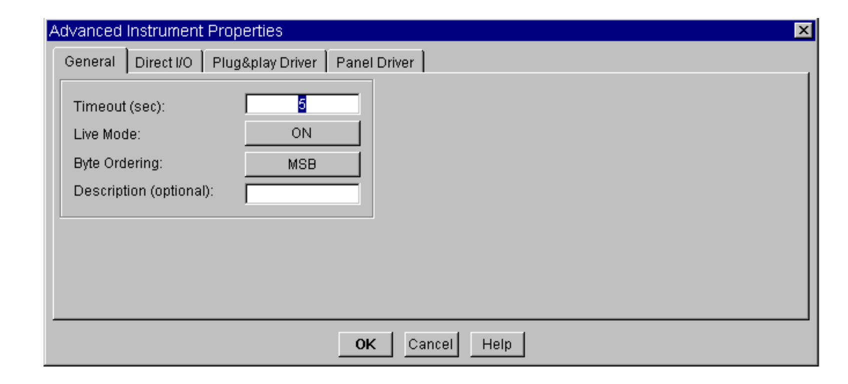


Abbildung 3-8. Das Dialogfenster "Advanced Instrument Properties"

Die Einträge im Ordner `General` lauten wie folgt:

Timeout Die maximale Anzahl von Sekunden, die für die Ausführung einer E/A-Transaktion zulässig ist, bis eine Fehlermeldung angezeigt wird.

Live Mode Gibt an, ob eine Echtzeit-Kommunikation mit dem Instrument erfolgt. Stellen Sie diese Angabe auf `OFF` ein, wenn kein Instrument angeschlossen ist. VEE verwendet den Standardwert `ON`.

Byte Ordering Gibt an, welche Reihenfolge das Gerät zum Lesen und Schreiben von Binärdaten verwendet. Das Feld schaltet um zwischen "Most Significant Byte First" (MSB) und "Least Significant Byte First". Alle IEEE488.2-kompatiblen Geräte müssen standardmäßig die MSB-Reihenfolge verwenden.

Description Geben Sie hier eine Beschreibung ein. Wenn beispielsweise die Instrumentnummer in der Titelleiste angezeigt werden soll, geben Sie hier diese Nummer ein.

4. Schalten Sie den `Live Mode` auf `OFF` um. Klicken Sie anschließend den Ordner `Panel Driver` an, um das Dialogfenster wie in Abbildung 3-9 dargestellt aufzurufen.

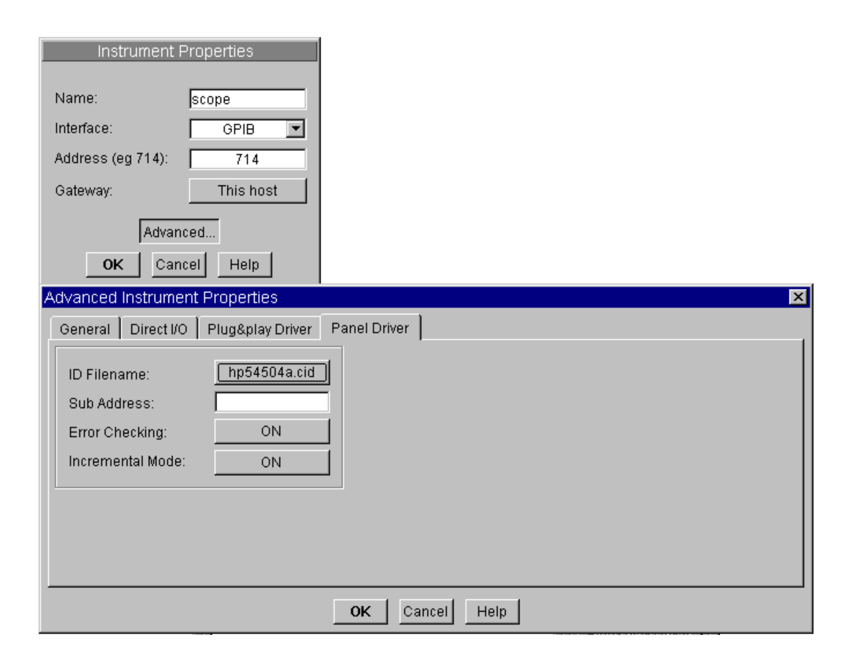


Abbildung 3-9. Der Ordner "Panel Driver"

5. Klicken Sie das Feld rechts von `ID Filename` an, um ein Listenfeld mit dem Namen `Read from what Instrument Driver?` zu öffnen. Die Liste enthält alle Panel-Treiberdateien, die mit Ihrer Version von VEE in dem angegebenen Verzeichnis gespeichert wurden.

Hinweis

Sie müssen die Panel-Treiber von der VEE-CD-ROM installiert haben, um das Beispiel ausführen zu können. Die `*.cid`-Dateien kennzeichnen die kompilierten Treiberdateien für das Instrument.

6. Blättern Sie in der Liste nach unten, um `hp54504a.cid` hervorzuheben, und klicken Sie anschließend `Open` an. In Abbildung 3-9 ist dieses Instrument bereits ausgewählt. Sie können auch eine hervorgehobene Datei doppelt anklicken, um sie auszuwählen.

Die weiteren Einträge im Ordner "Panel Driver" lauten wie folgt:

Sub Address Lassen Sie dieses Feld leer. Die Unteradresse wird nur von Nicht-VXI-Gehäuseinstrumenten zur Kennzeichnung von Zusatzmodulen verwendet.

Error Checking Behalten Sie die Standardeinstellung ON für die Fehlerprüfung bei. Die Fehlerprüfung kann für einen höheren Durchsatz ausgeschaltet werden; in diesem Fall wird jedoch nicht auf E/A-Fehler geprüft.

Incremental Mode Behalten Sie die Standardeinstellung ON bei. Der Inkrementelle Modus kann auch ausgeschaltet werden; in diesem Fall wird die gesamte Befehlsfolge für den Status des Instruments bei jeder Änderung einer Einstellung gesendet.

7. Klicken Sie OK an, um zum Fenster Instrument Properties zurückzukehren. Klicken Sie OK an.

   Die Liste der verfügbaren Instrumente sollte jetzt eine Instrumentenkonfiguration mit dem Namen scope enthalten, die die Treiberdatei hp54504a.cid verwendet, wie in Abbildung 3-10 gezeigt. Für das Instrument ist keine Busadresse angegeben, da es nicht tatsächlich vorhanden ist. Sie können das Programm in diesem Modus entwickeln und die Adresse später hinzufügen, wenn das Instrument bereit ist zum Anschließen an den Computer.

   Tip: Wenn Sie ein Feld ausgefüllt haben, drücken Sie die **Tabulatortaste**, um zum nächsten Feld zu springen. Mit der Tastenkombination **Umschalt-Tabulator** können Sie zum vorigen Feld wechseln. Das Drücken der **Eingabetaste** hat die gleiche Wirkung wie das Anklicken von OK. VEE schließt das Dialogfenster.

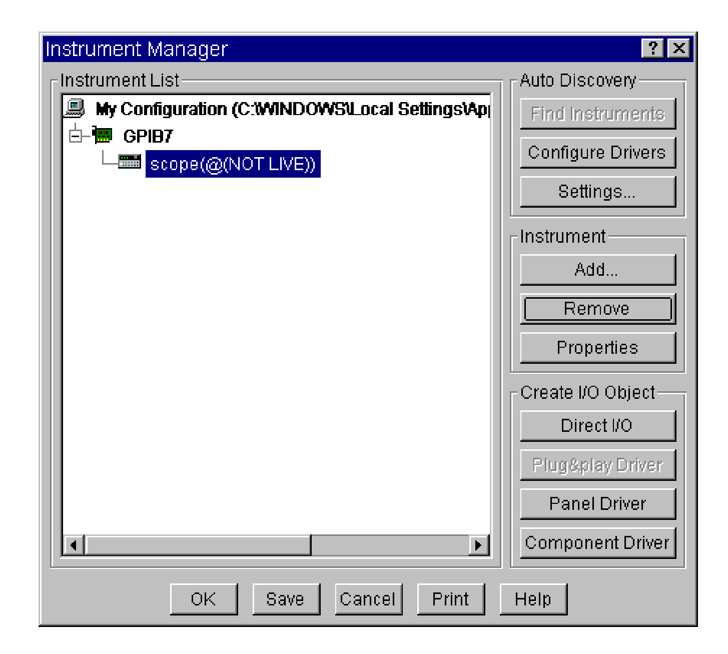


Abbildung 3-10. Oszilloskop zur Liste der Instrumente hinzugefügt

8. Klicken Sie `Save` an, um das Fenster `Instrument Manager` zu schließen. (Sie können auch `Panel Driver` unter `Create I/O Object` anklicken, um das Objekt sofort in das Programm einzufügen; VEE speichert in diesem Fall die Konfiguration automatisch.)

Sie haben jetzt das HP 54504A Oszilloskop mit dem Namen `scope` der Liste der Instrumente hinzugefügt. Sie können jetzt diesen Treiber bei der Programmierung verwenden, obwohl das eigentliche Instrument gar nicht vorhanden ist.

Auswählen eines Instruments zur Verwendung in einem Programm

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus.

2. Heben Sie die Auswahl `scope(@(NOT LIVE))` hervor und klicken Sie `Panel Driver` unter `Create I/O Object` an.

Hinweis

Im Instrument Manager können Sie häufig verschiedene Arten von Objekten unter Create I/O Object erstellen, je nachdem, welche Art von Instrument konfiguriert ist. Wenn Sie für diese Übung beispielsweise Direct I/O statt Panel Driver ausgewählt hatten, erhalten Sie ein Objekt Direct I/O mit dem Namen scope(@(NOT LIVE)). VEE bietet außerdem einen Component Driver, der eine Teilmenge der Funktionen eines Panel Driver verwendet. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

3. Positionieren Sie den Umriss des scope-Fensters, und klicken Sie mit der Maustaste, um das Fenster zu platzieren. Die Anzeige sollte jetzt aussehen wie in Abbildung 3-11.

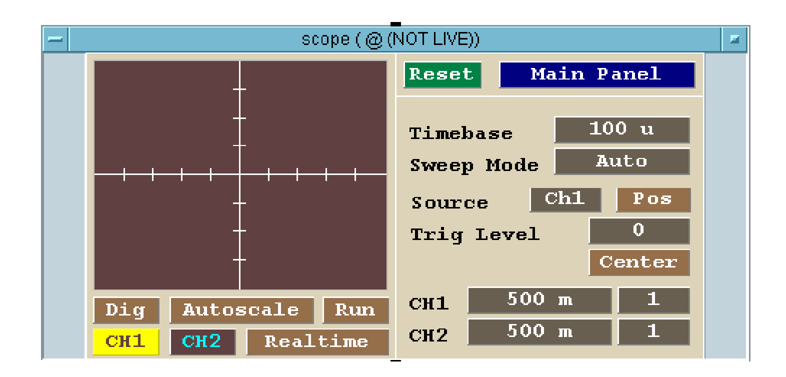


Abbildung 3-11. Auswählen von scope(@(NOT LIVE))

Sie können den Panel-Treiber jetzt wie jedes beliebige VEE-Objekt verwenden.

Hinzufügen des physischen Instruments zur Konfiguration

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus und heben Sie `scope` hervor. Klicken Sie `Properties` unter `Instrument...` an.

2. Klicken Sie doppelt auf das Feld `Address`, um den aktuellen Eintrag hervorzuheben, und geben Sie `709` ein. Die `7` in `709` gibt die logische Einheit an. (Wenn die logische Einheit GPIB (HP-IB) nicht 7 ist, ersetzen Sie die 7 durch die tatsächliche logische Nummer.) Die `9` in `709` ist die Standardadresse für Oszilloskope.

3. Klicken Sie `Advanced:` an und schalten Sie den `Live Mode` auf `ON`. Klicken Sie anschließend `OK` an. Klicken Sie `OK` an, um das Fenster `Instrument Properties` zu schließen.

4. Klicken Sie `Save` an, um die Änderungen zu speichern.

Verwenden eines Panel-Treibers

In dieser Übung wird der HP 3325B Funktionsgenerator als Beispiel verwendet. Das Prinzip der Verwendung ist für alle VEE-Panel-Treiber das gleiche. Durch die Verwendung eines Panel-Treibers statt der direkten Programmierung des Instruments sparen Sie viel Zeit bei der Entwicklung und der Änderung von Programmen. Änderungen an den Instrumenteinstellungen werden über eine Menüauswahl durchgeführt oder durch Ändern von Feldern in Dialogfenstern. Wenn das Instrument angeschlossen und der `Live Mode` auf `ON` gestellt ist, werden die von Ihnen vorgenommenen Änderungen im Instrument registriert.

Zur Verwendung eines Panel-Treibers in einem Programm fügen Sie Ein- und Ausgänge nach Bedarf hinzu und verbinden Sie den Panel-Treiber mit anderen Objekten. Sie können mehrere Exemplare eines Treibers in einem Programm verwenden, um verschiedene Statusvarianten des Instruments zu verwenden. In VEE können Sie einen Panel-Treiber als Symbol anzeigen, um Platz zu sparen, oder Sie können die offene Ansicht verwenden, um die Einstellungen des Instruments anzuzeigen. Sie können die Einstellungen auch ändern, während das Programm ausgeführt wird.

Übung 3-2: Ändern der Einstellungen eines Panel-Treibers

1. Wählen Sie `I/O ⇒ Instrument Manager...` aus. Wählen Sie `My Configuration` aus, klicken Sie `Add...` unter `Instrument` aus, um das Dialogfenster `Instrument Properties` anzuzeigen, und bearbeiten Sie die Informationen wie folgt:

Name	Ändern Sie den Namen in `fgen` und drücken Sie zweimal die **Tabulatortaste**, um zum Feld `Address` zu wechseln.
Address (Adresse)	Ändern Sie diesen Wert in `713` bzw. in die gewünschte Busadresse.

2. Klicken Sie `Advanced` im Ordner `General` an und schalten Sie den `Live Mode` auf `OFF`.

3. Klicken Sie den Ordner `Panel Driver` an und tragen Sie für `ID Filename:` den Wert `hp3325b.cid` ein. Klicken Sie zweimal `OK` an, um zum Fenster `Instrument Manager` zurückzukehren.

4. Klicken Sie `Panel Driver` unter `Create I/O Object` an. Platzieren Sie das Objekt auf der linken Seite des Arbeitsbereichs. (Dieser Prozess ist für alle Instrumente gleich, sofern das Instrument konfiguriert und der Liste hinzugefügt wurde.)

Hinweis

Sie programmieren, ohne dass das Instrument angeschlossen ist. Wäre das Instrument angeschlossen, würden Sie in der Konfiguration die richtige Adresse eintragen.

5. Klicken Sie `Sine` im Feld `Function` an, um ein Einblendmenü anzuzeigen, und wählen Sie in diesem Menü `Triangle` aus (siehe Abbildung 3-12).

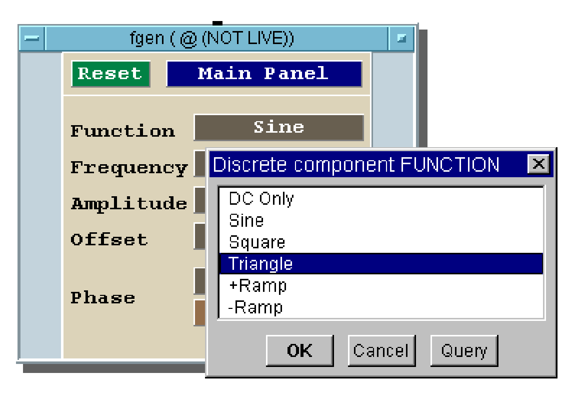


Abbildung 3-12. Das Einblendmenü Function zu fgen

6. Klicken Sie das Feld rechts von `Frequency` an.

7. Geben Sie `100` im Dialogfenster der "Continuous"-Komponente `FREQUENCY` ein und klicken Sie `OK` an. Beachten Sie, dass die Einstellung für `Frequency` geändert wurde.

Sie können die Einstellungen von Instrumenten bei allen Treibern auf die gleiche Weise ändern. Wenn das Instrument mit einer Adresse konfiguriert und der `Live Mode` auf `ON` eingestellt ist, werden alle im Treiberfenster vorgenommenen Änderungen von dem Instrument berücksichtigt.

Wechseln zu anderen Fenstern des gleichen Treibers

Die meisten Treiber enthalten mehrere Fenster, um die Benutzeroberfläche einfacher zu gestalten. Klicken Sie zum Wechseln zu einem anderen Fenster `Main Panel` im Objekt an, um ein Menü mit den verfügbaren Fenstern aufzurufen.

1. Klicken Sie im Objekt `Panel Driver` die Option `Main Panel` an und wählen Sie `Sweep` im Menü `Discrete Component` aus (siehe Abbildung 3-13).

2. Klicken Sie `OK` an, um das `Sweep Panel` anzuzeigen. Sie können auch die übrigen Fenster aufrufen, um sich einen Überblick über die verfügbaren Funktionen zu verschaffen.

3. Klicken Sie `OK` an, um zum `Main Panel` zurückzukehren.

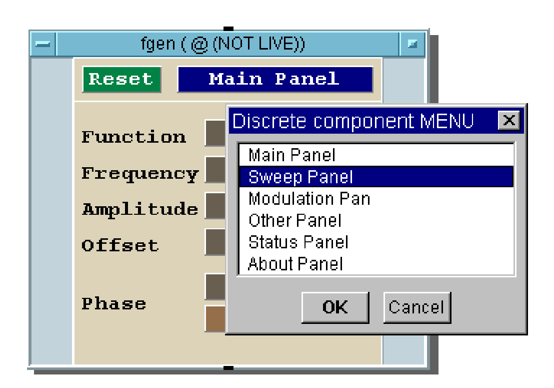


Abbildung 3-13. Einblendmenü "Sweep Panel" im Menü "Discrete Component"

Hinzufügen von Eingängen und/oder Ausgängen zu einem Panel-Treiber

Zusätzlich zu der direkten Interaktion über das Fenster können Sie die Einstellungen steuern oder Daten aus einem Instrument in ein Programm einlesen, indem Sie dem Treiber Datenein- oder -ausgänge hinzufügen. Die Ein- und Ausgangsbereiche sind in Abbildung 3-14 dargestellt.

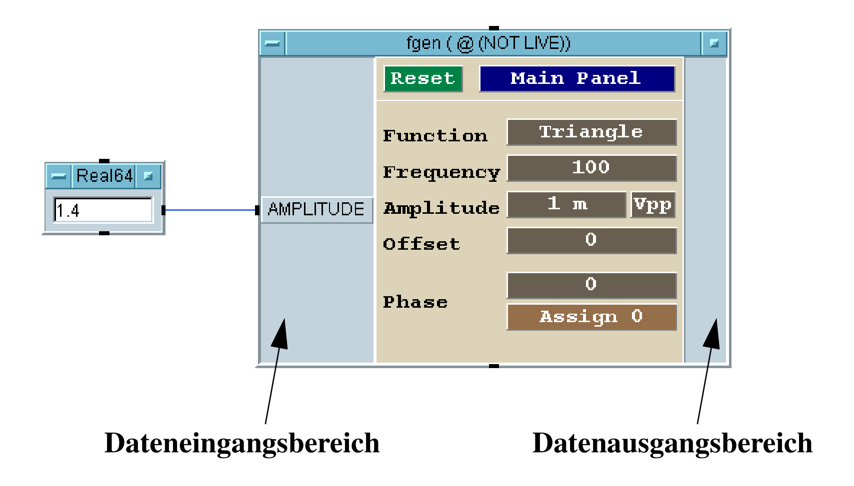


Abbildung 3-14. Die Datenein- und -Ausgangsbereiche für einen Treiber

1. Positionieren Sie den Mauszeiger auf den Dateineingangsbereich des Fensters für das Instrument "Funktionsgenerator", und drücken Sie **Strg-A**, um einen Dateneingabeanschluss hinzuzufügen. Daraufhin wird eine Liste der Instrumentkomponenten angezeigt.

2. Wählen Sie die gewünschte Komponente in dem angezeigten Menü aus.

Hinweis

Sie können auch das Objektmenü öffnen und `Add Terminal` by `Component ⇒ Select Input Component` auswählen. Wählen Sie anschließend das gewünschte Komponentenfeld am Treiber aus.

Verwenden Sie diesen Prozess auch zum Hinzufügen eines Datenausgangs, indem Sie den Mauszeiger auf den Datenausgangsbereich positionieren.

Löschen von Dateneingangs- oder -ausgangsanschlüssen

Positionieren Sie den Mauszeiger auf den Anschluss und drücken Sie **Strg-D**.

Hinweis

Sie können auch das Objektmenü öffnen, `Delete Terminal` ⇒ `Input...` auswählen und den gewünschten Eingang in dem angezeigten Menü auswählen.

Eigene Experimente

Geben Sie einen Status für den Funktionsgenerator HP 3325B oder einen beliebigen anderen Funktionsgenerator an. Ändern Sie die Einstellung `Function` auf eine `Square`-Kurve. Fügen Sie Eingangskomponenten für `Amplitude` und `Frequency` hinzu. Erstellen Sie Eingabe-Dialogfenster für die Amplitude und die Frequenz, und ändern Sie die Titel in eine Eingabeaufforderung für den Bediener. Geben Sie verschiedene Werte für die Amplitude und die Frequenz ein, und führen Sie das Programm aus, um festzustellen, ob die Bedienereingaben durch die Einstellungen geändert wurden. (Wenn ein Instrument angeschlossen ist, werden die Einstellungen geändert, sofern der `Live Mode` auf `ON` eingestellt ist.)

Verwenden des Objekts Direct I/O

Wenn für ein bestimmtes Instrument kein Treiber verfügbar ist oder Sie einen höheren Durchsatz erzielen wollen, verwenden Sie das Objekt "Direct I/O".

Übung 3-3: Verwenden von Direct I/O

In dieser Übung konfigurieren Sie den Funktionsgenerator HP 3325B mit Hilfe des Objekts "Direct I/O".

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus.

2. Heben Sie den Eintrag `fgen(@(NOT LIVE))` hervor und wählen Sie `Instrument` ⇒ `Properties` aus.

3. Klicken Sie `Advanced` an.Wählen Sie den Ordner `Direct I/O` aus, wie in Abbildung 3-15 gezeigt. Sehen Sie sich die verfügbaren Optionen an, und klicken Sie `OK` an, um zu `Instrument Properties` zurückzukehren. Klicken Sie jetzt erneut `OK` an, um zum `Instrument Manager` zurückzukehren.

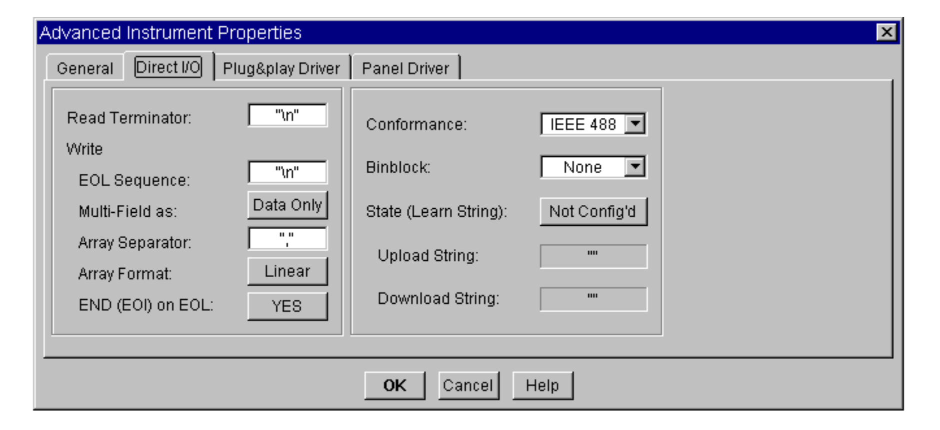


Abbildung 3-15. Der Konfigurationsordner "Direct I/O"

Hinweis

In diesem Beispiel wird die GPIB-Schnittstelle (IEEE488) verwendet. Informationen zum Konfigurieren von seriellen, GPIO- oder VXI-Instrumenten finden Sie im Handbuch *VEE Pro Advanced Techniques*.

4. Vergewissern Sie sich zum Platzieren des Objekts auf dem Bildschirm, dass `fgen(@(NOT LIVE))` noch immer hervorgehoben ist, und klicken Sie `Create I/O Object` ⇒ `Direct I/O` an. Abbildung 3-16 zeigt das Objekt "Direct I/O".

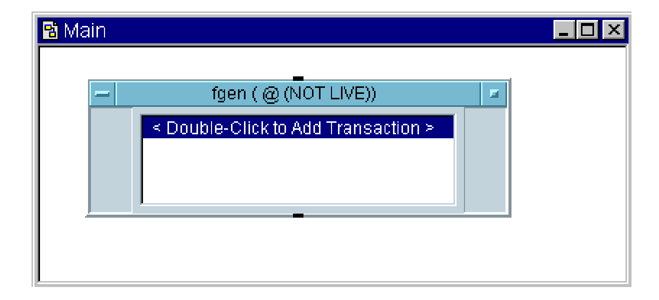


Abbildung 3-16. Ein Objekt "Direct I/O"

Zur Verwendung eines Objekts `Direct I/O` in einem Programm müssen Sie E/A-Transaktionen konfigurieren. Im nächsten Abschnitt werden das Schreiben von Textbefehlen, das Lesen von Daten und das Hoch- und Herunterladen des Instrumentenstatus beschrieben.

Senden eines einzelnen Textbefehls an ein Instrument

Geben Sie zum Senden eines einzelnen Textbefehls an ein Instrument die entsprechende Zeichenfolge ein. Die meisten GPIB-Instrumente verwenden alphanumerische Zeichenfolgen für Befehle, die an das Instrument gesendet werden. Wenn Sie beispielsweise an den Funktionsgenerator HP3325B einen Befehl senden wollen, der die Amplitude auf 5 Volt setzt, geben Sie die Befehlszeichenfolge `"AM 5 VO"` ein.

In dieser Übung wird der Funktionsgenerator HP 3325B verwendet, der im vorigen Abschnitt konfiguriert wurde. Gehen Sie ggf. zurück zum Abschnitt “Verwenden des Objekts Direct I/O” auf Seite 153 und konfigurieren Sie das Instrument, bevor Sie mit dieser Übung fortfahren.

1. Klicken Sie im Objekt `fgen(@(NOT LIVE))` die Transaktionsleiste doppelt an, um das Dialogfenster `I/O Transaction` aufzurufen (siehe Abbildung 3-17).

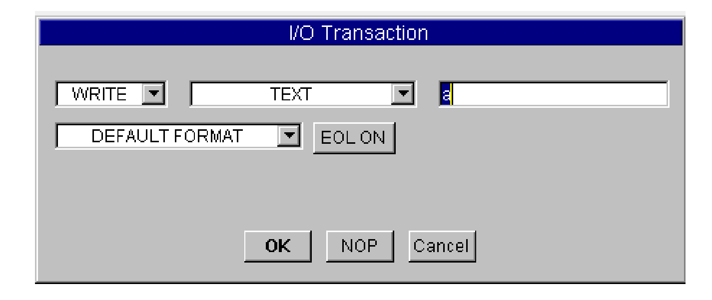


Abbildung 3-17. Das Dialogfenster "I/O Transaction"

Der Abwärtspfeil neben `WRITE` zeigt ein Menü mit Transaktionen an: `READ` (LESEN), `WRITE` (SCHREIBEN), `EXECUTE` (AUSFÜHREN) und `WAIT` (WARTEN). Verwenden Sie zum Schreiben von Daten auf ein Instrument die Standardauswahl. Öffnen Sie das Objektmenü und schlagen Sie Informationen zu den einzelnen Aktionen unter `Help` nach.

2. Verwenden Sie die Standardauswahl `WRITE`, `TEXT`, `DEFAULT FORMAT` und `EOL ON`. Klicken Sie das Eingabefeld `a` an, geben Sie `"AM 5 VO"` (mit den Anführungszeichen) ein und klicken Sie `OK` an.

Jetzt sollte die Transaktion `WRITE TEXT "AM 5 VO" EOL` angezeigt werden wie in Abbildung 3-18 dargestellt. Der Text in Anführungszeichen ist der Befehl, der bei der Ausführung des Programms an das Instrument HP3325B gesendet wird.

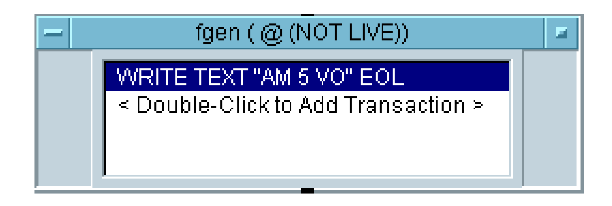


Abbildung 3-18. Eine Transaktion "Direct I/O"

In den meisten Fällen ist dieser Prozess beim Senden von Befehlen an Instrumente gleich. Es gibt jedoch auch Instrumente, für die am Ende jedes gesendeten Befehls oder jeder gesendeten Befehlsgruppe bestimmte Zeichen angegeben werden müssen. Sie finden diese Informationen in der Dokumentation zu dem jeweiligen Instrument. Geben Sie die entsprechenden Zeichen im Dialogfenster `Direct I/O Configuration` an.

Senden einer Ausdrucksliste an ein Instrument

In manchen Fällen ist es erforderlich, eine Liste von Ausdrücken an ein Instrument zu senden. Es könnte beispielsweise sein, dass im `Funktionsgenerator` eine Reihe von Frequenzen durchlaufen werden soll. Dies kann mit einer "Direct I/O"-Transaktion erreicht werden; Sie verwenden dabei eine Variable für die Frequenz in einer Ausdrucksliste und fügen dem Objekt "Direct I/O" einen Dateneingang für diese Variable hinzu. In den folgenden Schritten wird beschrieben, wie Sie eine Ausdrucksliste an ein Instrument senden.

1. Platzieren Sie ein zweites Objekt `Direct I/O` für den HP3325B im Hauptfenster. Klicken Sie doppelt im Transaktionsbereich, um das Dialogfenster `I/O Transaction` aufzurufen.

   Sie können alle Standardwerte mit Ausnahme der Befehlszeichenfolge verwenden. Verwenden Sie in diesem Fall das Format `"FR"`, <*frequency*>, `"HZ"`. Dies ist eine Ausdrucksliste; die einzelnen Ausdrücke sind durch Kommas abgegrenzt. Die Frequenz wird durch die Variable `A` dargestellt; diese Variable wird als Dateneingang für das Objekt `Direct I/O` verwendet.

2. Klicken Sie das Eingabefeld für Befehlszeichenfolgen ("Command string") an und geben Sie ein: `"FR",A,"HZ"`. (Wenn `A` beispielsweise `100` ist, sendet VEE die Zeichenfolge `"FR100HZ"`.) Klicken Sie `OK` an. Beachten Sie, dass VEE automatisch einen Dateneingangs-Pin mit der Bezeichnung A hinzufügt.

3. Wählen Sie `Flow ⇒ Repeat ⇒ For Range` aus und platzieren Sie das Objekt links von dem Objekt `Direct I/O`.

4. Verbinden Sie den Datenausgangs-Pin `For Range` mit dem Dateneingangs-Pin von `Direct I/O`.

5. Ändern Sie die Felder in `For Range` wie folgt: `From 10`, `Thru 1.8M` und `Step 50k`.

   `For Range` sendet jetzt Zahlen von 10 bis 1,8 Millionen in Intervallschritten zu 50.000. Wenn die Zahlen vom Objekt "Direct I/O" empfangen werden, bewirkt die Befehlszeichenfolge, dass der Funktionsgenerator die entsprechenden Frequenzen ausgibt. Die Konfiguration des Objekts "Direct I/O" sollte aussehen wie in Abbildung 3-19.

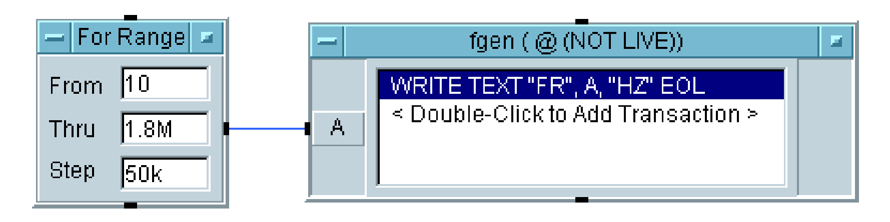


Abbildung 3-19. Konfiguration von "Direct I/O" mit einer Eingangsvariablen

6. (Optional) Verbinden Sie - falls verfügbar - einen HP3325B mit Ihrem Computer und ändern Sie die Konfiguration dieses Objekts "Direct I/O" so, dass sie die Adresse des Instruments enthält. Führen Sie das Programm aus, und Sie sehen, wie das Instrument diese Frequenzen generiert.

Einlesen von Daten von einem Instrument

Instrumente senden Daten in vielen verschiedenen Formaten an Ihren Computer. Zum Einlesen von Daten von einem Instrument müssen Sie wissen, welchen Datentyp Sie einlesen wollen, und ob die zurückzugebenden Daten einen einzelnen Wert (ein Skalar) oder einen Array bilden sollen. Außerdem müssen Sie wissen, ob das Instrument Daten als Text (ASCII) oder als Binärdaten zurückgibt.

Sie finden diese Informationen in der Dokumentation zu dem jeweiligen Instrument, oder Sie können mit dem VEE `Bus I/O Monitor` im Menü

`I/O` die zurückgegebenen Daten prüfen. Diese Informationen legen fest, wie die E/A-Transaktion konfiguriert werden muss.

In diesem Beispiel wird ein HP3478A Multimeter an den in der vorigen Übung beschriebenen Funktionsgenerator HP3325B angeschlossen. Wenn der Generator eine bestimmte Frequenz aussendet, löst der Multimeter ein Lesen aus und sendet das Ergebnis zurück an VEE. Die folgenden Schritte beschreiben, wie die Transaktion für den Multimeter konfiguriert wird.

Hinweis

In diesem Beispiel wird eine Transaktion `READ TEXT` beschrieben. Weitere Auswahloptionen `READ` sind `BINARY`, `BINBLOCK` und `CONTAINER`; diese werden im Handbuch *VEE Pro Advanced Techniques* ausführlich beschrieben.

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus. Klicken Sie `Add...` an. Ändern Sie den Namen in `dvm`. Klicken Sie `Advanced...` an und stellen Sie den `Live Mode:` auf `OFF` ein. Sofern Sie keinen HP3478A angeschlossen haben, klicken Sie `OK` an, um zum Instrument Manager zurückzukehren. (Falls Sie *doch* einen HP3478A angeschlossen haben, ändern Sie die Adresse entsprechend, damit das Instrument die Befehle protokollieren kann.)

2. Heben Sie die Auswahl `dvm(@(NOT LIVE))` hervor und klicken Sie `Direct I/O` unter `Create I/O Object` an.

3. Klicken Sie doppelt die Leiste `<Double-Click to Add Transaction>` an, um das Dialogfenster `I/O Transaction` anzuzeigen.

4. Heben Sie das Eingabefeld hervor, geben Sie `"T5"` ein und klicken Sie `OK` an. Dadurch wird der Befehl `"T"` an das Instrument geschrieben. `T5` ist der Befehl für ein einzelnes Auslösen an den Multimeter.

5. Öffnen Sie das Objektmenü und klicken Sie `Add Trans...` an, um eine weitere Transaktionsleiste hinzuzufügen, oder verwenden Sie `<Double-Click to Add Transaction>` zum Anzeigen einer Transaktion und zum Anzeigen des Dialogfensters `I/O Transaction`.

6. Klicken Sie den Abwärtspfeil neben `WRITE` an, um ein Dropdown-Menü aufzurufen, und wählen Sie `READ` aus. Wenn Sie `READ` auswählen, werden neue Schaltflächen im Fenster `I/O Transaction` angezeigt.

7. Überprüfen Sie das Eingabefeld `ExpressionList`, um sicherzustellen, dass es ein `x` enthält. Drücken Sie die **Tabulatortaste**, um zum nächsten Feld zu springen. Die von einem Instrument zurückgegebenen Daten werden an die Datenausgangs-Pins gesendet. In diesem Fall werden Daten von dem Instrument eingelesen und an einen Datenausgang mit dem Namen `x` gesendet.

Hinweis Bei den Namen wird *nicht* zwischen Groß- und Kleinbuchstaben unterschieden.

8. Behalten Sie den Standardwert `REAL64 FORMAT` bei. Der Multimeter gibt einzelne Lesevorgänge als reale Zahlen zurück.

9. Behalten Sie `DEFAULT NUM CHARS` bei.

   Der Standardwert für die Anzahl der Zeichen beträgt 20. Wenn Sie diese Anzahl ändern wollen, klicken Sie `DEFAULT NUM CHARS` an, um auf `MAX NUM CHARS` umzuschalten, und ändern Sie die Anzahl `20` in die gewünschte Anzahl.

10. Behalten Sie die Angabe `SCALAR` bei und klicken Sie `OK` an.

    Die Transaktion wird in der Leiste als `READ TEXT X REAL64` angezeigt. Beachten Sie, dass VEE automatisch einen Datenausgang mit dem Namen `x` hinzufügt.

Hinweis Wenn das Instrument einen Array von Werten zurückgibt, klicken Sie das Menü `SCALAR` im Dialogfenster `I/O Transaction` an, um das Menü für verschiedene Abmessungen aufzurufen (siehe Abbildung 3-20). Wenn Sie die Array-Abmessung ausgewählt haben, müssen Sie auch eine Größe für den Array angeben.

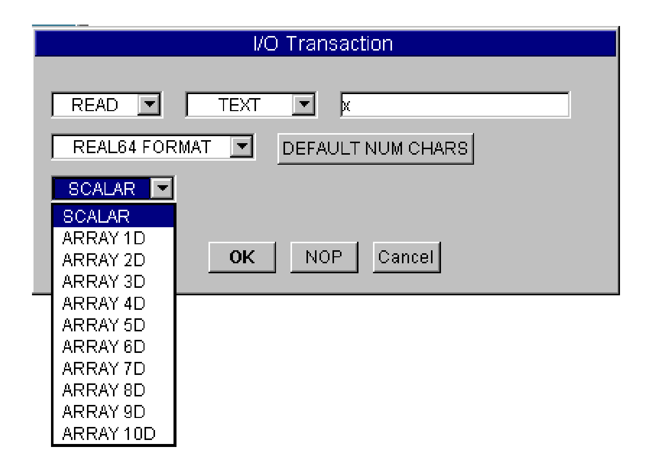


Abbildung 3-20. Konfigurieren einer READ-Transaktion

11. Fügen Sie ein Objekt `Display` ⇒ `AlphaNumeric` auf der rechten Seite hinzu und verbinden Sie seinen Eingang mit dem `Direct I/O`-Ausgang mit der Beschriftung `X`.

    Die beiden Transaktionen "Direct I/O" sollten aussehen wie in Abbildung 3-21.

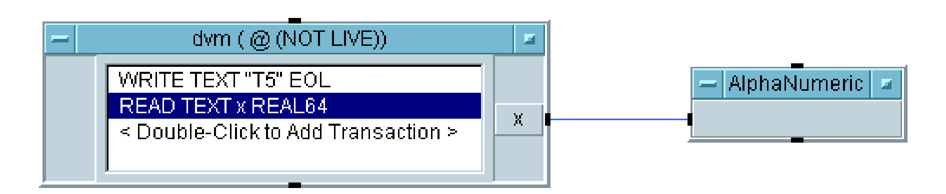


Abbildung 3-21. Direct I/O konfiguriert zum Einlesen einer Messung

Der Prozess zum Konfigurieren einer Transaktion ist sehr ähnlich, unabhängig von dem Datenformat für die Transaktion `READ TEXT`. Sie können auch die weiteren verfügbaren Formate anzeigen. Ausführlichere Informationen zu den einzelnen Elementen finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Zum Erstellen eines vollständigen Testprogramms können dieses Multimeter-Objekt und ein Funktionsgenerator-Objekt mit VEE-Daten- und -Anzeigeobjekten kombiniert werden. Voll funktionsfähige Testprogramme sind in VEE sehr einfach zu erstellen. Es würde jedoch den Rahmen dieses Einführungskapitels sprengen, ausführliche Details zu allen verschiedenen verfügbaren Instrumenten aufzuzeigen. Ausführliche komplexe Beispiele finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Hochladen und Herunterladen von Statusangaben zu Instrumenten

Manche Instrumente bieten eine “Learn-String”-Funktion. Diese Lernzeichenfolge umfasst alle Funktionseinstellungen, aus denen sich der Status eines Instruments zusammensetzt. Das Objekt "Direct I/O" lädt diese Lernzeichenfolge hoch und speichert sie mit diesem speziellen "Direct I/O"-Objekt; später können Sie sie in das Instrument in dem Programm herunterladen. Führen Sie zum Hochladen eines Instrumentenstatus die folgenden Schritte aus:

1. Stellen Sie das Instrument manuell auf den gewünschten Status ein.

2. Öffnen Sie das `Direct I/O`-Objektmenü und klicken Sie `Upload State` an.

   Dieser Status ist jetzt diesem speziellen Exemplar des Objekts "Direct I/O" zugeordnet.

3. Öffnen Sie ein Dialogfenster `I/O Transaction` durch Doppelklicken im Transaktionsbereich.

4. Klicken Sie `TEXT` an, wählen Sie `STATE (LEARN STRING)` aus und klicken Sie anschließend `OK` an, um das Fenster `I/O Transaction` zu schließen. Der zuvor erfasste Status wird an das Instrument gesendet, wenn diese `WRITE`-Transaktion ausgeführt wird.

Das Hoch- und Herunterladen wird über die Einstellungen im Dialogfenster "Direct I/O Configuration" gesteuert. Sofern die Kompatibilität zu IEEE 488.2 gegeben ist, verarbeitet VEE automatisch die Lernzeichenfolgen über die 488.2 *LRN? Definition. Bei einer Kompatibilität zu IEEE 488 gibt

`Upload String` den Befehl zum Abfragen des Status an, und `Download String` gibt den Befehl an, der beim Herunterladen der Statuszeichenfolge vorangestellt wird. Abbildung 3-22 zeigt hierzu ein Beispiel.

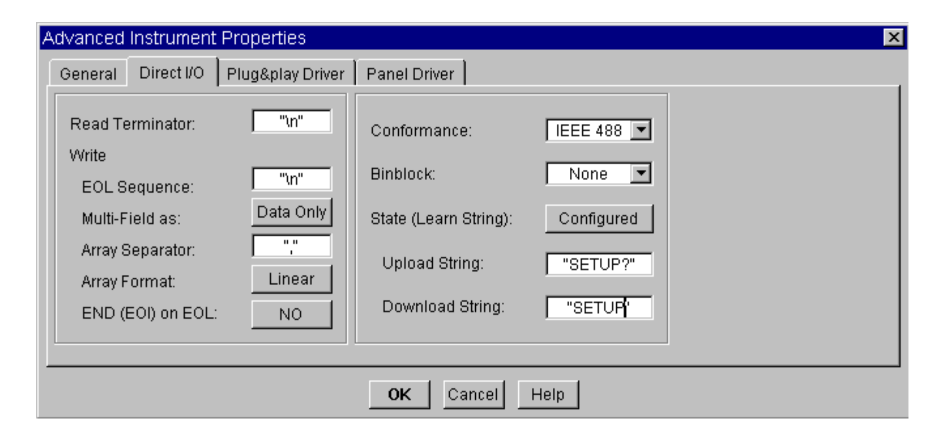


Abbildung 3-22. Lernzeichenfolge-Konfiguration für HP54100A

Die Kompatibilität kann definiert sein für IEEE 488 oder IEEE 488.2. In diesem Beispiel wird das HP 54100A Digitalisierungs-Oszilloskop verwendet, das IEEE 488 entspricht und einen `"SETUP?"` erfordert, um die Lernzeichenfolge abzufragen; beim Herunterladen muss der Lernzeichenfolge `"SETUP"` vorangestellt werden. Bei Auswahl von `Configured for State (Learn String)` erscheinen zwei weitere Felder mit den Bezeichnungen `Upload String` und `Download String`. Die entsprechenden Zeichenfolgen wurden in den Eingabefeldern eingetragen.

Verwenden von PC-Zusatzkarten

VEE bietet drei verschiedene Möglichkeiten zum Steuern von PC-Zusatzkarten:

1. ODAS-Treiber vom Hersteller der PC-Zusatzkarte.

2. Data Translation Visual Programming Interface. (Bestellen Sie die VPI-Anwendung direkt bei Data Translation.)

3. Dynamic Link Libraries vom Hersteller der PC-Zusatzkarte, z. B. ComputerBoards oder Meilhaus. (Im Abschnitt “Verwenden der Dynamic Link Libraries” auf Seite 440 finden Sie Informationen zur Verwendung von Dynamic Link Libraries.)

Verwenden von ODAS-Treibern

Gehen Sie zum Installieren der PC-Zusatzkarte und der ODAS-Treibersoftware sowie zum Ausführen des ODAS-Konfigurationsprogramms anhand der Anleitungen des Herstellers vor. Konfigurieren Sie anschließend den Treiber in VEE.

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus. Wählen Sie `Find Instruments` aus. Der `Instrument Manager` zeigt Einträge wie die in Abbildung 3-23 dargestellten an.

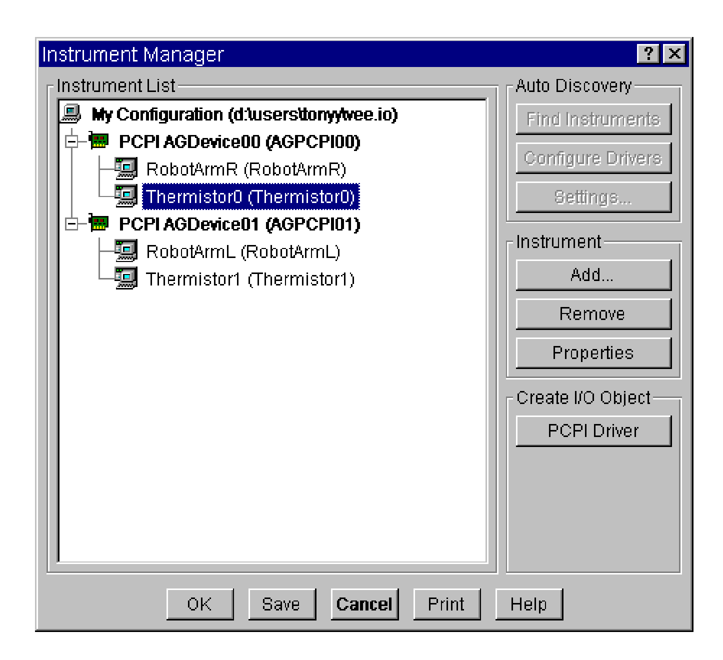


Abbildung 3-23. ODAS-Treibereinträge im Instrument Manager

2. Wählen Sie einen der Untereinträge wie `Thermistor0` aus, und wählen Sie unter `Create I/O Object` die Option `PCPI Driver` aus. Klicken Sie mit der Maustaste, um das Objekt zu platzieren. Es erscheint als `Formula`-Objekt in VEE (siehe Abbildung 3-24).

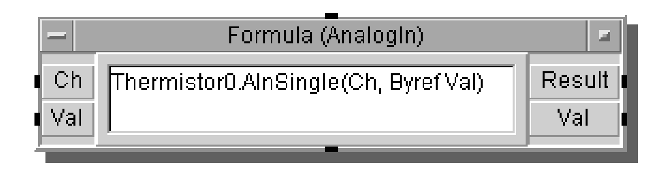


Abbildung 3-24. PC-Zusatzkarte mit ODAS-Treiber als Formelobjekt

Ausführliche Informationen zum Verwenden von PC-Zusatzkarten mit ODAS-Treibern finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Data Translation Visual Programming Interface (VPI).

Data Translation VPI kann zusammen mit VEE für ein reibungsloses Erfassen von Daten über PC-Zusatzkarten verwendet werden. Durch die Flexibilität der Data Translation Open Layers Standards haben Sie Zugriff auf mehr als 50 Datenerfassungskarten.

VPI arbeitet direkt mit Datenerfassungskarten nach ISA-, PCI- und USB-Standards; diese Karten erfordern nur eine geringe Channel-Kapazität. VPI fügt VEE eine Menüauswahl hinzu und ein Symbol für eine spezifische PC-Zusatzkarte zur Datenerfassung. Diese Elemente steuern die Data Translation Hardware-Funktionalität.

Amplicon

Amplicon bietet eine breite Palette von analogen und digitalen E/A-PC-Zusatzkarten der Serie 200; alle diese Karten unterstützen VEE.

Die Software-Schnittstelle ist Teil des Amplicon AmpDIO-Treiberpakets, einer 32-Bit-API mit einer Multithread-DLL für Windows und Unterstützung einer interrupt-gesteuerten Erfassung. Die API umfasst mehr als 100 Aufrufe für eine effiziente und flexible Programmierung als kompilierte Funktion über eine VEE-spezifische Definitionsdatei und der Möglichkeit, bis zu acht Karten in einem Programm zu verwenden.

Zusätzlich zu der Amplicon Produktfamilie von Zusatzkarten einschließlich der seriellen Kommunikationsgeräte kann Amplicon auch Karten für die Datenerfassung, die serielle Kommunikation oder GPIB-Anwendungen von einer Vielzahl anderer Hersteller liefern.

Abbildung 3-25 zeigt die VEE-Laufzeit-Software (im Lieferumfang der Amplicon analogen Ausgangskarten PCI224 und PCI234 sowie der analogen Eingangskarten PCI230 und PCI260 enthalten) für gleichzeitige Ein- und Ausgangssignale an einem PC.

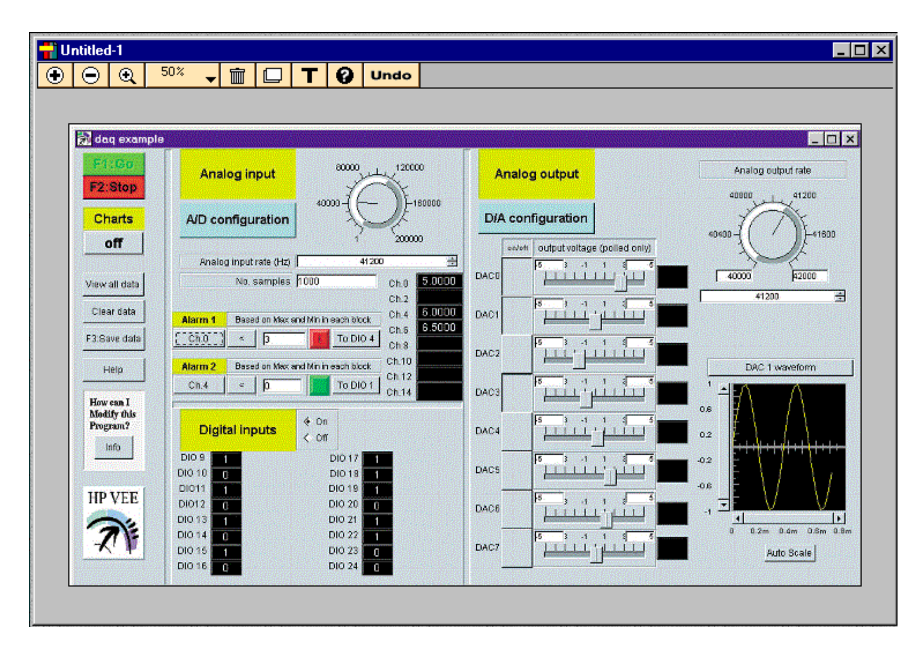


Abbildung 3-25. Amplicon Datenerfassung Beispiel

ComputerBoards PC-Zusatzkarten

ComputerBoards bietet preisgünstige und leistungsstarke PC-Zusatzkarten, die mit VEE kompatibel sind. (Eine vollständige Liste der Hersteller unterstützter PC-Zusatzkarten finden Sie in der VEE-Literatur oder in *VEE Pro Advanced Techniques*.)

Sie installieren einfach die Karte und die dazugehörige E/A-Bibliothek und konfigurieren die Karte mit einem vom Hersteller bereitgestellten Programm. Gehen Sie zum Anschließen der Karte an das Gerät anhand der Anleitungen vom Hersteller vor. Importieren Sie die Bibliothek in VEE. Anschließend können Sie die Messungsfunktionen aus der ComputerBoards E/A-Bibliothek aufrufen. Die nachfolgenden Abbildungen zeigen ein vom Hersteller bereitgestelltes Demonstrationsprogramm.

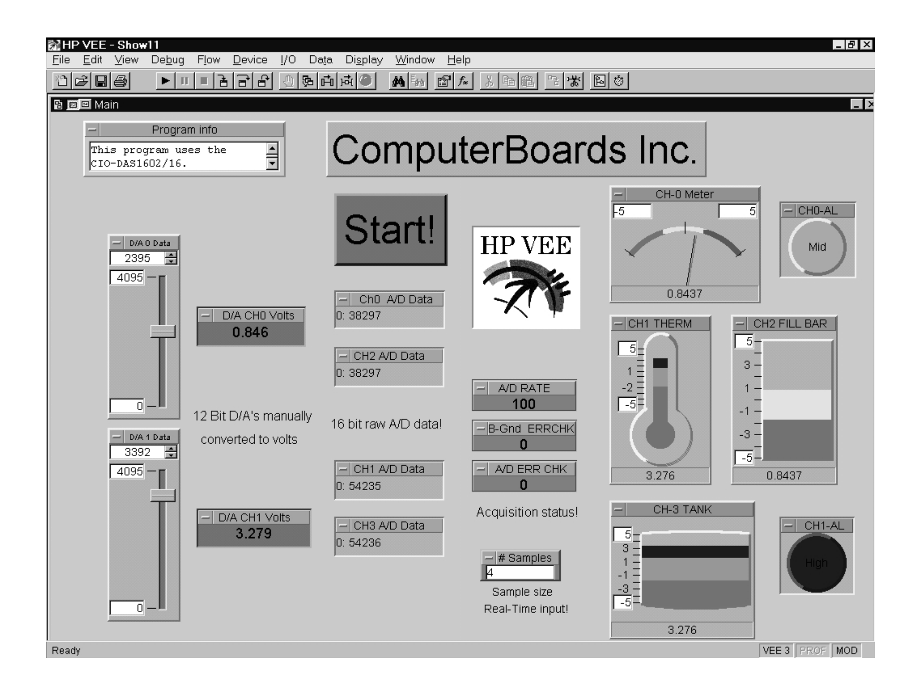


Abbildung 3-26. VEE Verwenden einer ComputerBoards 100 KHz-Karte

Abbildung 3-26 zeigt die Fensteransicht des Demonstrationsprogramms mit der 100 KHz A/D-Karte. Abbildung 3-27 zeigt VEE beim Importieren der ComputerBoards E/A-Bibliothek, die diese Funktionsaufrufe zur Datenerfassung möglich macht.

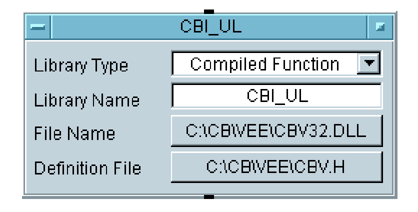


Abbildung 3-27. Importieren der ComputerBoards E/A-Bibliothek

Meilhaus Electronic ME-DriverSystem

Meilhaus Electronic ist eines der führenden Unternehmen in Europa im Bereich der Entwicklung, Fertigung und dem Vertrieb von PC-Datenerfassungs- und -Schnittstellenlösungen. Das ME-DriverSystem für Windows auf CD-ROM ist im Lieferumfang aller von Meilhaus Electronic hergestellten Datenerfassungskarten (d.h. der ME Serie) enthalten. Das ME-DriverSystem ist ebenfalls in die VEE-Menüstruktur integriert.

Nach der Installation des ME-DriverSystem für VEE erscheinen die Treiberfunktionen in einem VEE-Pulldown-Menü. Abbildung 3-28 zeigt das Menü "ME Board" in VEE.

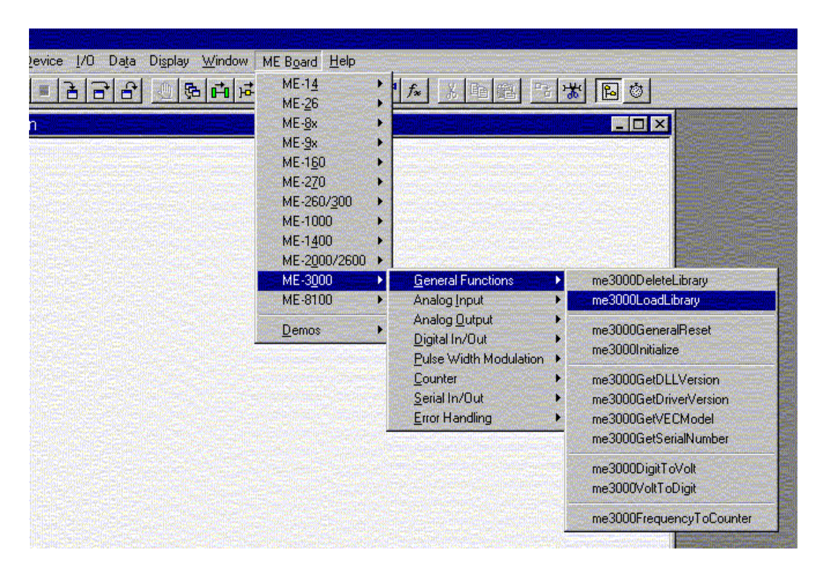


Abbildung 3-28. ME Board-Menü in VEE

Die zweite Menüebene bietet funktionale Gruppen wie den analogen Eingang und Ausgang, digitale E/A und spezielle Funktionen bestimmter Karten. Abbildung 3-29 zeigt das Benutzerfenster für die Datenerfassungskarte ME-3000.

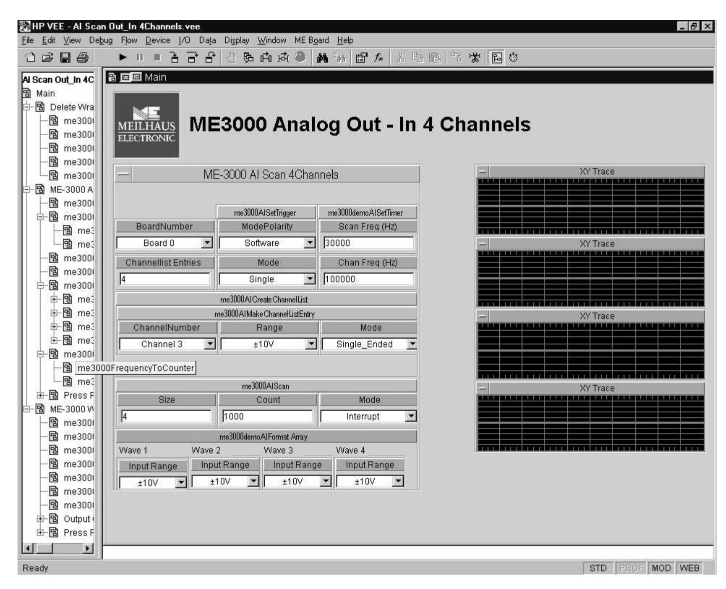


Abbildung 3-29. Benutzerfenster für die Datenerfassungskarte ME-3000

Im dritten Menü schließlich befinden sich die eigentlichen Funktionen, z. B. me3000AISingle. Abbildung 3-30 zeigt das Funktionsfenster.

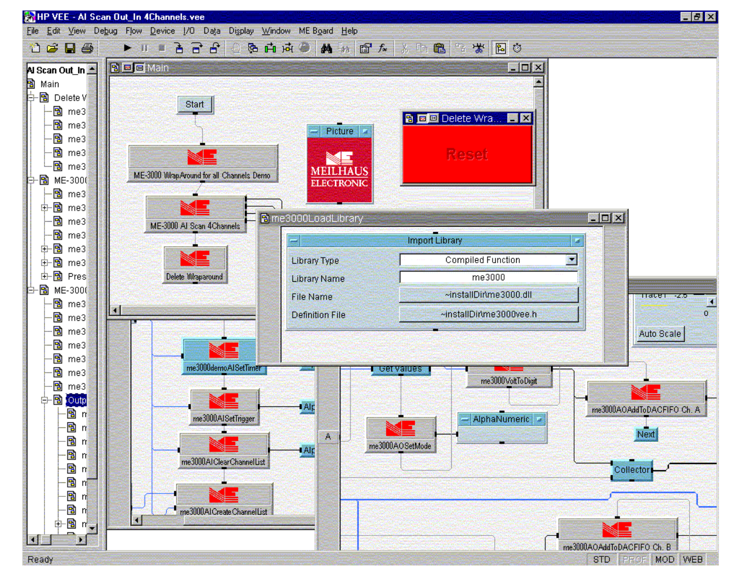


Abbildung 3-30. Funktionsfenster für ME-DriverSystem

Verwenden eines VXI*Plug&Play*-Treibers

VXI*Plug&Play*-Treiber werden von verschiedenen Instrumentenherstellern herausgegeben und unterstützt. Hierbei handelt es sich um C-basierende Treiber, die für eine maximale Leistung und Bedienerfreundlichkeit konzipiert wurden.

Agilent VEE ist vollständig VXI*Plug&Play*-kompatibel. Alle verfügbaren VXI*Plug&Play*-Treiber von Agilent Technologies werden als separate Produkte geliefert und sind auch im Web unter `http://www.agilent.com/find/inst_drivers` verfügbar. Die gleichen Treiber sind auch in VEE zusammen mit allen Agilent Technologies Panel-Treibern enthalten. Zum Abrufen der VXI*Plug&Play*-Treiber für weitere Instrumente wenden Sie sich an den jeweiligen Hersteller des Instruments.

Übung 3-4: Konfigurieren eines VXI*Plug&Play*-Treibers

Dieses Beispiel beschreibt die Konfiguration eines HPE1412-Treibers.

1. Wählen Sie `I/O` ⇒ `Instrument Manager...` aus.

2. Heben Sie `My configuration` hervor und klicken Sie `Add...` unter `Instrument` an, um das Dialogfenster `Instrument Properties` aufzurufen. Geben Sie einen Namen ein, z. B. `Instrument`, und klicken Sie `Advanced...` an, um das Dialogfenster `Advanced Instrument Properties` anzuzeigen.

3. Schalten Sie im Dialogfenster `Advanced Instrument Properties` den `Live Mode:` auf `OFF` und wählen Sie den Ordner `Plug&play Driver` aus. Klicken Sie das Feld `Plug&play Driver Name:` an, um das Dropdown-Menü anzuzeigen, das alle auf dem Computer installierten Treiber anzeigt. Dieses Beispiel verwendet den Treiber HPE1412 (siehe Abbildung 3-31).

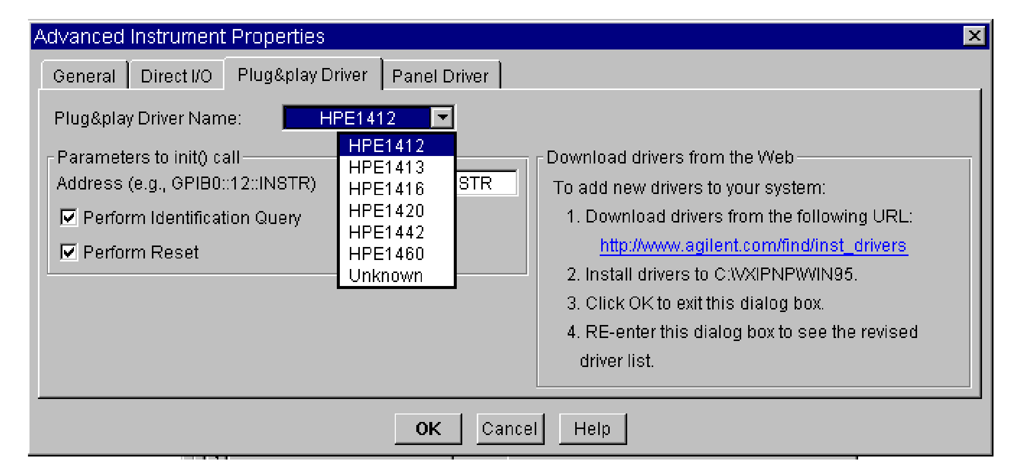


Abbildung 3-31. Auswählen eines VXI*Plug&Play*-Treibers

4. Wählen Sie den Treiber `HPE1412` aus, klicken Sie `OK` an, um zum Dialogfenster `Instrument Properties` zurückzukehren, und klicken `OK` an, um zum `Instrument Manager` zurückzukehren. Es sollte jetzt ein Eintrag angezeigt werden für `Instrument(@(NOT LIVE))`.

5. Heben Sie die Auswahl `Instrument(@(NOT LIVE))` hervor und wählen Sie `Plug&play Driver` unter `Create I/O Object` aus. Klicken Sie mit der Maustaste, um das Objekt zu platzieren.

Hinweis

In VEE ähnelt ein VXI*Plug&Play*-Treiber einem "Direct I/O"-Objekt.

Wenn Sie mit dem Instrument Messungen durchführen wollen, müssen Sie im VXI*Plug&Play*-Treiber E/A-Transaktionen konfigurieren, die C-Funktionen verwenden. Der Treiber bietet Fenster, in denen Sie die gewünschten Funktionen auswählen können.

6. Klicken Sie doppelt auf die Transaktionsleiste mit der Beschriftung `<Double-click to Add Function>` und wählen Sie `Select a Function Panel` aus, wie in Abbildung 3-32 gezeigt. Abbildung 3-33 zeigt die Hierarchie der Funktionen im Funktionsfenster. Beachten Sie, dass `Help` für das ausgewählte Element in dem Dialogfenster angezeigt wird.

Hinweis

VEE initialisiert das Instrument automatisch. Sie brauchen keine `init`-Funktion zu verwenden wie in anderen Sprachen.

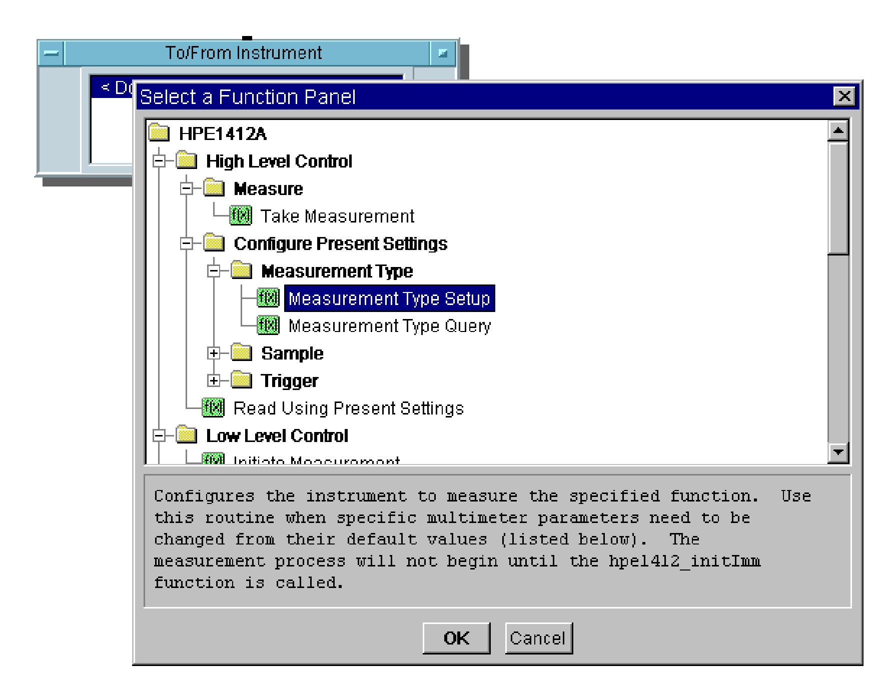


Abbildung 3-32. Auswählen einer Funktion für einen VXI*Plug&Play*-Treiber

7. Klicken Sie `Configure Present Settings` ⇒ `Measurement Type` ⇒ `Measurement Type Setup` an. Das Fenster `Edit Function` wird angezeigt. Klicken Sie unter `func`, um die Dropdown-Liste anzuzeigen. Wählen Sie den Standardwert `DC Voltage` (Gleichspannung) aus (siehe Abbildung 3-33).

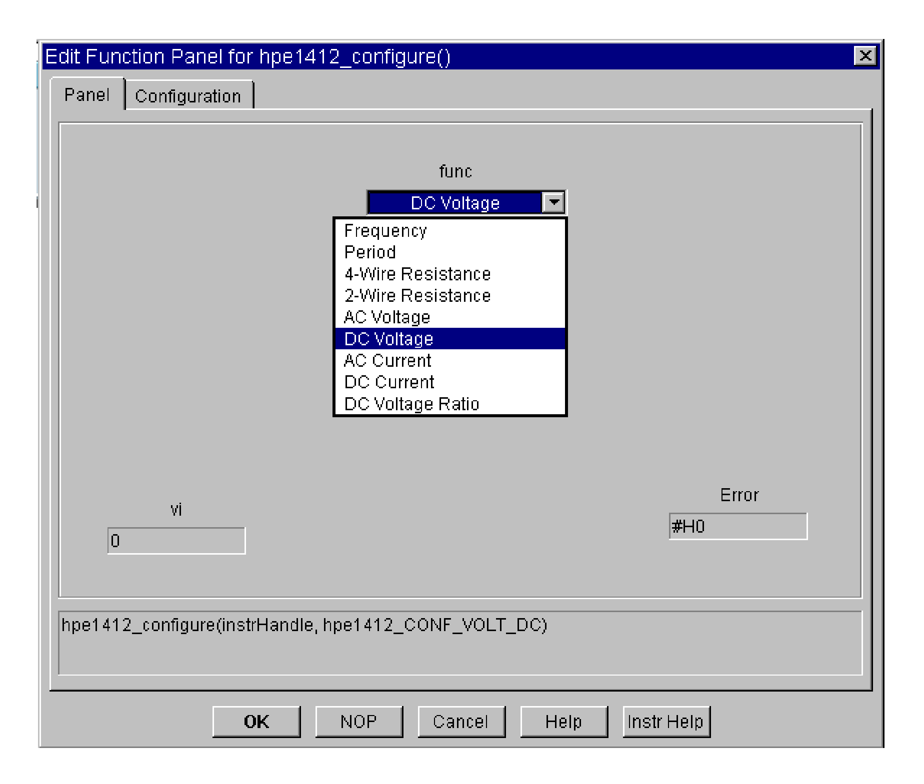


Abbildung 3-33. Das Fenster "HPE1412 Edit Function"

8. Klicken Sie `OK` an. Das Objekt `To/From Instrument` enthält jetzt einen Eintrag für `hpe1412_configure(instruHandle,hpe1412_CONF_VOLT_DC)`, wie in Abbildung 3-34 gezeigt.

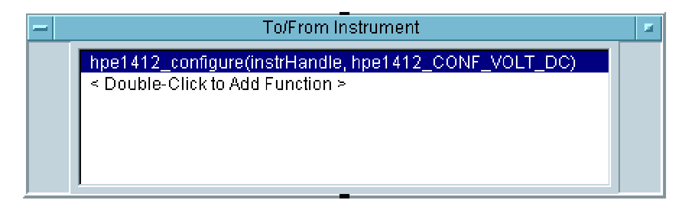


Abbildung 3-34. Funktion "DC Voltage" in VXIPlug&Play-Objekt

9. Klicken Sie im Objekt `To/From Instrument` doppelt, um eine Funktion hinzuzufügen, und wählen Sie `Take Measurement` unter `Measure` aus. Klicken Sie den Ordner `Configuration` an, um das in Abbildung 3-35 dargestellte Dialogfenster aufzurufen.

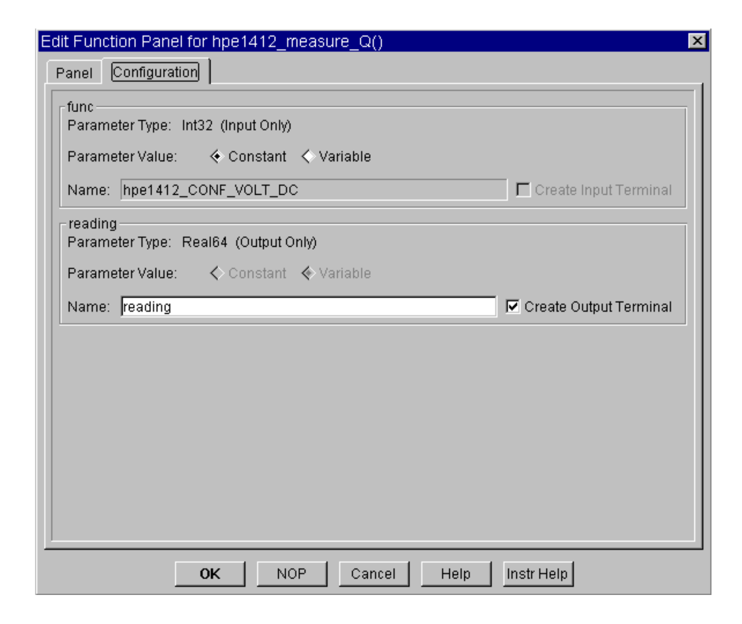


Abbildung 3-35. Konfigurationsordner im Fenster "Edit Function"

10. Klicken Sie `OK` an. Im Objekt `To/From Instrument` wird ein zweiter Funktionsaufruf aufgelistet (siehe Abbildung 3-36).

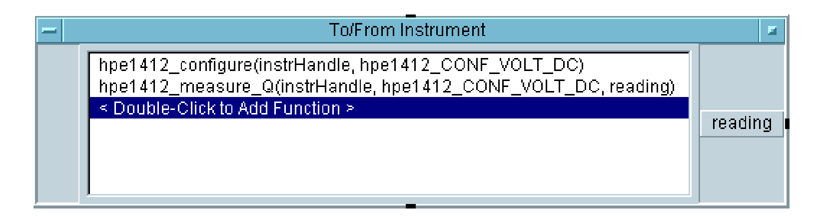


Abbildung 3-36. HPE1412-Treiber bereit zum Einlesen DC

Weitere E/A-Funktionen

- Lernen Sie die ganze Leistung der VEE-E/A-Funktionen im Untermenü `I/O ⇒ Advanced I/O` kennen: `Interface Operations`, `Instrument Event`, `Interface Event` und `MultiInstrument Direct I/O`.

- Mit "Bus I/O Monitor" im I/O-Menü können Sie die Busaktivität anzeigen, drucken oder zur Fehlerbehebung speichern.

- VEE enthält einen ActiveX-Automatisierungs-Server zur programmgesteuerten Suche von Instrumenten. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

- Sie können die E/A-Konfiguration auch zur Ausführungszeit programmgesteuert ändern. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Erläutern der Vorteile von Instrumententreibern und des Objekts "Direct I/O".
- Erläutern des Prozesses zur Steuerung von Instrumenten.
- Konfigurieren eines Instruments für einen Treiber.
- Konfigurieren eines Instruments für das Objekt "Direct I/O".
- Ändern der Einstellungen an einem Instrumententreiber.
- Hinzufügen und Löschen von Komponentenein- und -ausgängen.
- Wechseln zu den verschiedenen Fenstern für einen Instrumententreiber.
- Verwenden des Objekts "Direct I/O" zum Schreiben von Befehlen an ein Instrument.
- Verwenden des Objekts "Direct I/O" zum Lesen von Daten aus einem Instrument.
- Hochladen und Herunterladen des Instrumentenstatus über Lernzeichen-folgen.
- Verwenden von VXI*Plug&Play*-Treibern zur Kommunikation mit einem Instrument.
- Erläutern der drei Methoden zum Steuern von PC-Zusatzkarten.

4

Analysieren und Anzeigen von Testdaten

Analysieren und Anzeigen von Testdaten

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- VEE-Datentypen
- VEE-Analysefunktionen
- Verwenden von mathematischen Objekten
- Verwenden des Formelobjekts
- Verwenden des MATLAB Script-Objekts
- VEE-Anzeigefunktionen
- Anpassen von Anzeigen

Erforderliche Zeit für dieses Kapitel: 1,5 Stunden

Überblick

In diesem Kapitel erfahren Sie mehr über die Analyse- und Anzeigefunktionen von VEE. Sie lernen, wie Sie die richtigen mathematischen Objekte für Ihre Anwendungen finden und wie Sie die Testergebnisse anzeigen, sodass Sie Daten schnell und einfach in hilfreiche Informationen umsetzen können.

Mithilfe der ActiveX-Automatisierung können Sie auch andere gängige Anwendungen wie MS Excel zum Analysieren der Daten verwenden. (Weitere Informationen hierzu finden Sie in Kapitel 6 “Einfaches Erstellen von Berichten mit ActiveX,” auf Seite 259.) Mit ActiveX-Steuerelementen können Sie auch VEE-externe Anzeigefunktionen verwenden. (Weitere Informationen hierzu finden Sie im Abschnitt “Verwenden eines ActiveX-Steuerelements” auf Seite 417). Der Schwerpunkt dieses Kapitels liegt auf den VEE-spezifischen Tools und dem Objekt "MATLAB Script", das zusammen mit VEE ausgeliefert wird.

Agilent VEE Datenformen und Datentypen

In einem VEE-Programm werden Daten entlang der Linien zwischen Objekten übertragen und von den nachfolgenden Objekten verarbeitet. Zum Angeben einer Gruppe von Daten packt VEE diese Daten in einen Container, der eine *Datenform* (*scalar* oder *array*) und einen *Datentyp* (z. B. `Int32`, `Real64` oder `Text`) hat.

Datenform: Ein *scalar* ist eine einzelne Zahl einschließlich komplexer Zahlen, die aus zwei oder mehr Komponenten bestehen. Ein *Array* enthält eine Gruppe von Datenelementen, die als eindimensionaler Array (Array 1D), zweidimensionaler Array (Array 2D) etc. angegeben werden können.

Datentypen: Die VEE-Datentypen sind in Tabelle 4-1 beschrieben.

Im Allgemeinen brauchen Sie sich um Datentypen oder -formen nicht zu kümmern, da die meisten Objekte mit beliebigen VEE-Datentypen arbeiten und Daten automatisch in den für das Objekt erforderlichen Typ umwandeln. Wenn beispielsweise die Anzeige eines Größenspektrums einen Wellenform-Datentyp empfängt, führt VEE automatisch eine Fast Fourier-Transformation durch, um die Daten aus dem Zeitbereich in den Frequenzbereich umzuwandeln.

Gelegentlich erwartet ein Objekt jedoch einen bestimmten Datentyp; sie sollten sich daher über die Datentypen im Klaren sein. Außerdem sollten Sie über die Unterschiede zwischen den von VEE und MATLAB unterstützten Datentypen Bescheid wissen. (Weitere Informationen hierzu finden Sie im Abschnitt “Arbeiten mit Datentypen” auf Seite 198.)

Die folgende Tabelle enthält einen kurzen Überblick über die VEE-Datentypen. Spezifische Punkte zu diesen Datentypen finden Sie in den nachfolgenden Kapiteln.

Tabelle 4-1. Agilent VEE Datentypen

Datentyp	Beschreibung
UInt8	Byte ohne Vorzeichen 0 bis 255.
Int16	Ein ganzzahliges 16-Bit-Zweierkomplement (-32768 bis 32767).

Tabelle 4-1. Agilent VEE Datentypen (Fortsetzung)

Datentyp	Beschreibung
Int32	Ein ganzzahliges 32-Bit-Zweierkomplement (-2147483648 to 2147483647).
Real32	Eine 32-Bit-Gleitkommazahl entsprechend dem Standard IEEE 754 (+/-3.40282347E+/-38).
Real64	Eine 64-Bit-Gleitkommazahl entsprechend dem Standard IEEE 754 (+/- 1.797693138623157 E308).
PComplex	Eine Größen- und Phasenkomponente in der Form (mag, @phase). Die Phase ist standardmäßig auf degrees (Grad) gesetzt, kann jedoch mit der Einstellung ⇒ Default Preferences ⇒ Trig Mode auf radians (Radiant) oder gradians (Gradient) gesetzt werden.
Complex	Eine rechtwinklige oder kartesische komplexe Zahl mit einer realen und einer imaginären Komponente in der Form (real, imag). Jede Komponente entspricht einem Real64. Die komplexe Zahl 1 + 2i wird beispielsweise als (1,2) dargestellt.
Waveform	Ein zusammengesetzter Datentyp aus Zeitbereichswerten, der die Real64-Werte gleichmäßig verteilter, linearer Punkte und die gesamte Zeitspanne der Wellenform enthält. Die Datenform einer Waveform muss ein eindimensionaler Array (Array 1D) sein.
Spectrum	Ein zusammengesetzter Datentyp aus Frequenzbereichswerten, der die PComplex-Werte von Punkten sowie die Minimum- und Maximumwerte der Frequenz enthält. Die Bereichsdaten können als Aufzeichnung oder linear zugeordnet werden. Die Datenform eines Spectrum muss ein eindimensionaler Array (Array 1D) sein.

Tabelle 4-1. Agilent VEE Datentypen (Fortsetzung)

Datentyp	Beschreibung
Coord	Ein zusammengesetzter Datentyp, der mindestens zwei Komponenten in der Form (x,y,...) enthält. Jede Komponente entspricht einem Real64. Die Datenform einer Coord muss ein Scalar oder ein Array 1D sein.
Enum	Eine Textzeichenfolge mit einem zugeordneten ganzzahligen Wert. Sie können auf die Ganzzahl mit der Funktion ordinal(x) zugreifen.
Text	Eine Folge aus alphanumerischen Zeichen.
Record	Ein zusammengesetzter Datentyp mit einem Feld für jeden Datentyp. Jedes Feld hat einen Namen und einen Container, der einen beliebigen Typ und eine beliebige Form (einschließlich Record) haben kann.
Object	Nur verwendet für ActiveX-Automatisierungs und -Steuerelemente, eine Referenz auf ein ActiveX-Steuerelement oder eine von einem Automatisierungs-Aufruf zurückgegebene Referenz. Genau genommen ist dies eine Referenz auf eine IDispatch- oder eine IUnknown-Schnittstelle.
Variant	Nur für ActiveX-Automatisierung und -Steuerelemente verwendet. Ein Datentyp, der für manche ActiveX-Methodenaufrufe als Parametertyp "By Ref" erforderlich ist.



Hinweis Sehen Sie Informationen zur gemeinsamen Nutzung von Daten in Mischumgebungen unter `I/O` ⇒ `To/From Socket` nach.

Agilent VEE-Analysefunktionen

VEE unterstützt allgemeine mathematische Operationen und Hunderte von weiteren Funktionen. Darüber hinaus enthält VEE auch die Funktion "MATLAB Script". Die Funktion "MATLAB Script" ist eine Untermenge des Standard-Vollprodukts MATLAB von The MathWorks. Sie bietet mathematische Funktionen in VEE einschließlich der Signalverarbeitung, der höheren Mathematik, der Datenanalyse sowie wissenschaftliche und Engineering-Grafik. Die MATLAB Script-Funktion ist mit VEE vollständig integriert, und Sie können MATLAB Script-Objekt in jedes VEE-Programm einbeziehen.

Wenn die erforderliche mathematische Funktion weder in VEE noch in MATLAB enthalten ist, stehen Ihnen verschiedene Möglichkeiten zur Verfügung. Sie können die Funktion mit dem Objekt `Formula` erstellen; dieses Objekt wird weiter hinten in diesem Kapitel beschrieben. Sie können die Funktion auch in einer Compilersprache wie beispielsweise C schreiben und mit VEE verknüpfen, oder Sie können die Kommunikation mit einer anderen Software-Anwendung von VEE aus verwenden.

Verwenden integrierter mathematischer Objekte

Unter VEE `Device ⇒ Function & Object Browser` können Sie auf integrierte (vorprogrammierte) mathematische Ausdrücke für VEE und MATLAB zugreifen.

Aufrufen integrierter Operatoren oder Funktionen

Wählen Sie zum Aufrufen mathematischer VEE-Operatoren und Funktionen `Device ⇒ Function & Object Browser` aus. Wenn Sie beispielsweise eine Formel erstellen wollen, die eine Zufallszahl in einem angegebenen Bereich zurückgibt, wählen Sie `Type: Built-in Functions`, `Category: Probability & Statistics` und `Functions: random` aus (siehe Abbildung 4-1).

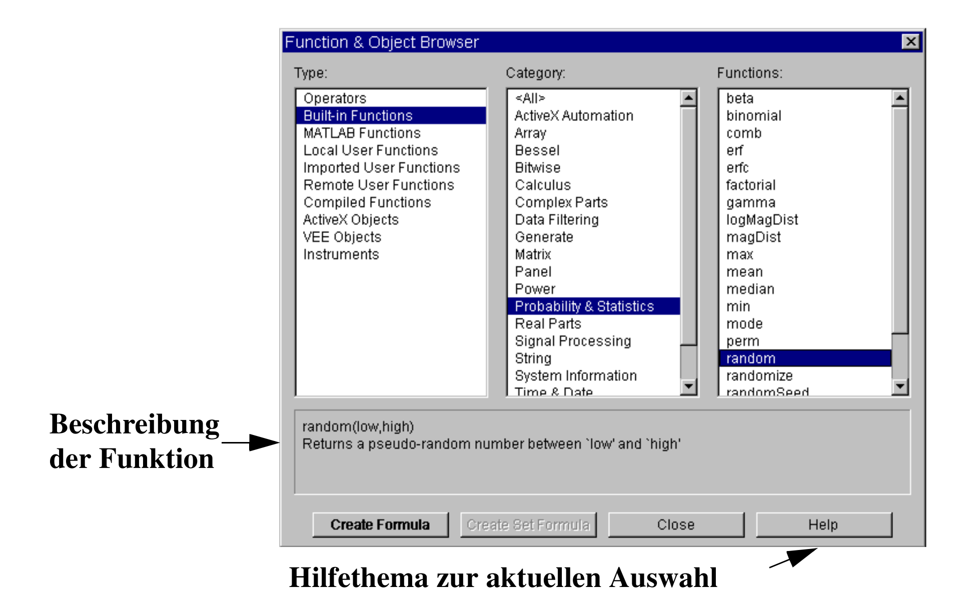


Abbildung 4-1. Eine VEE-Funktion im Function & Object Browser

Beachten Sie, dass der `Function & Object Browser` eine kurze Beschreibung der aktuellen Auswahl anzeigt, wie in Abbildung 4-1 dargestellt. Sie können auch die Schaltfläche `Hilfe` anklicken, um eine ausführlichere Beschreibung der aktuellen Auswahl und Informationen zur Definition, Verwendung, Syntax sowie Beispiele aufzurufen.

Wählen Sie zum Aufrufen von MATLAB-Operatoren und -Funktionen `Device ⇒ Function & Object Browser` und unter `Type:` die Option `MATLAB Functions` aus. Um beispielsweise Wurzeln in Polynome umzuwandeln, wählen Sie `Type: MATLAB Functions`, `Category: Interpolation & Polynomials`, and `Functions: poly` aus (siehe Abbildung 4-2).

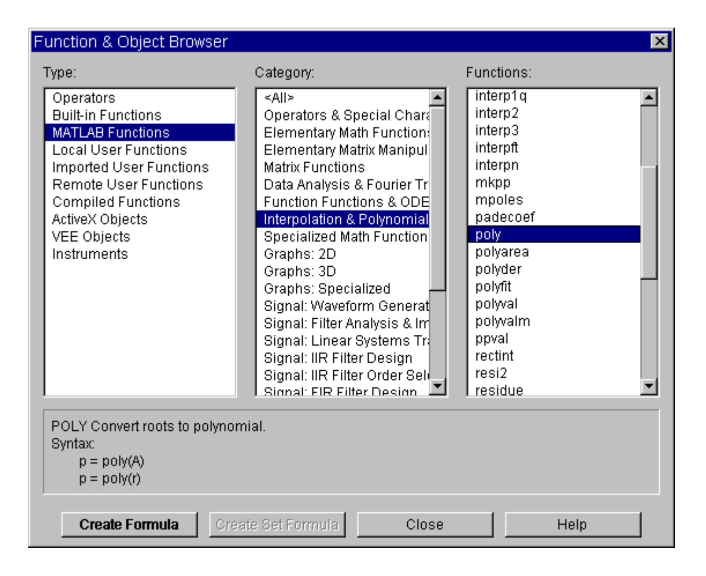


Abbildung 4-2. Eine MATLAB-Funktion im Function & Object Browser

Auch hier wird eine kurze Beschreibung der aktuellen Auswahl im `Function & Object Browser` angezeigt; durch Anklicken von `Help` werden weitere Beschreibungen zur aktuellen Auswahl angezeigt. Die "MATLAB Runtime Engine and Script" wird im Abschnitt “Verwenden von MATLAB Script in Agilent VEE” auf Seite 194 ausführlich beschrieben.

Übung 4-1: Berechnen der Standardabweichung

Generieren Sie eine Kosinus-Wellenform mit einer Frequenz von 1 kHz, einer Amplitude von 1 V und einem Zeitbereich von 20 ms, dargestellt durch 256 Punkte. Berechnen Sie seine Standardabweichung und zeigen Sie diese an.

1. Wählen Sie `Device ⇒ Virtual Source ⇒ Function Generator` aus. Stellen Sie die `Frequency` entsprechend ein und stellen Sie sie als Symbol dar.

2. Wählen Sie `Device ⇒ Function & Object Browser` und anschließend `Built-in Functions`, `Probability & Statistics`, und `sdev` aus. Klicken Sie `Create Formula` an.

Hinweis

Sie können direkt zum Dialogfenster "Function & Object Browser" wechseln, indem Sie das Symbol ***fx*** in der Symbolleiste anklicken (siehe Abbildung 4-3) oder **Strg-I** drücken.



Abbildung 4-3. Öffnen des Function and Object Browser über das Symbol "fx"

3. Öffnen Sie das Objektmenü für `sdev()`, um Hilfeinformationen anzuzeigen.

Hinweis

Das Objekt `sdev(x)` ist definiert als die Quadratwurzel der Varianz von `x`, und `x` kann den Typ UInt8, Int16, Int32, Real32, Real64, Coord oder Waveform haben. Der Funktionsgenerator gibt den Datentyp Waveform aus.

4. Verbinden Sie den `Function Generator` mit `sdev(x)`.

5. Wählen Sie `Display` ⇒ `AlphaNumeric` aus und verbinden Sie das Objekt mit dem Datenausgangs-Pin von `sdev(x)`.

6. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 4-4.

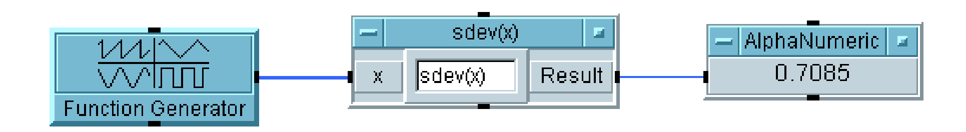


Abbildung 4-4. Berechnen der Standardabweichung

Erstellen von Ausdrücken mit dem Formelobjekt

Mit dem `Formula`-Objekt können mathematische Ausdrücke in VEE geschrieben werden. Die Variablen in dem Ausdruck sind die Namen der Dateneingangs-Pins oder globale Variablen. Das Ergebnis der Auswertung des Ausdrucks wird am Datenausgangs-Pin angelegt.

Abbildung 4-5 zeigt ein `Formula`-Objekt. Das Eingangsfeld des Ausdrucks befindet sich in der Mitte des Objekts. Ein Standardausdruck `(2*A+3)` kennzeichnet die Stelle für die Eingabe der Formel. Klicken Sie das Feld doppelt an, um einen anderen Ausdruck einzugeben

Hinweis

Sie können einen `Formula`-Ausdruck auch über mehrere Zeilen eingeben. Wenn ein `Formula`-Ausdruck ein **Zeilenschaltungszeichen** enthält, wird der Ausdruck als mehrzeiliger Einzelausdruck interpretiert. Wenn eine `Formula` Anweisungen enthält, die durch Semikolons (`;`) getrennt sind, werden diese Anweisungen als mehrere Ausdrücke in der Formel interpretiert.

Sie können auch Standard-Bearbeitungsbefehle zum Bearbeiten von Ausdrücken in einer Formel verwenden. Sie können beispielsweise die Maus ziehen, um Zeichen hervorzuheben, diese Zeichen mit **Strg-C** kopieren, mit **Strg-V** einfügen oder mit **Strg-X** löschen.

Eingabefeld

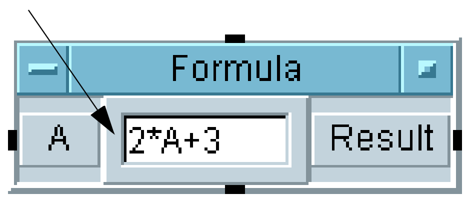


Abbildung 4-5. Das Formelobjekt

Hinweis

Alle aus dem Typ `Devices ⇒ Function & Object Browser Built-in` erstellten Typen sind einfach Formeln, deren Ausdrücke bereits richtig gesetzt wurden. Sie können auch geändert werden, um Funktionen zu kombinieren und Eingänge hinzuzufügen oder zu löschen. Sie können auch mehrzeilige Einträge im `Formula`-Objekt vornehmen und den Ausgangsanschlüssen Werte zuweisen.

Auswerten eines Ausdrucks mit dem Formelobjekt

In diesem Beispiel werten Sie den Ausdruck `2*A^6-B` aus, wobei `A=2` und `B=1`. (Beachten Sie das Zeichen ^ für die Potenzierung.)

Hinweis

Bei den Variablennamen wird *nicht* zwischen Groß- und Kleinschreibung unterschieden.

1. Wählen Sie `Device ⇒ Formula` aus. Klicken Sie das Eingabefeld `Formula` an und geben Sie `2*A^6-B` ein.

2. Positionieren Sie den Mauszeiger auf den Dateineingangsbereich (aber nicht direkt auf dem Eingang `A`) und drücken Sie **Strg-A**, um einen Eingangs-Pin hinzuzufügen.

Hinweis

Dieser Pin wird standardmäßig als `B` beschriftet; Sie können ihn jedoch auch umbenennen.

3. Wählen Sie `Data ⇒ Constant ⇒ Int32` aus, klonen Sie das Objekt durch Auswählen von `Clone` im Objektmenü, und verbinden Sie die beiden Int32-Objekte zu den Formeleingängen `A` und `B`.

4. Geben Sie `2` im Eingabefenster `A Int32` und `1` im Eingabefenster `B Int32` ein.

5. Wählen Sie `Display ⇒ AlphaNumeric` aus, verbinden Sie das Objekt mit dem Ausgang der Formel und führen Sie das Programm aus. Es sollte das Ergebnis 127 anzeigen, wie in Abbildung 4-6 dargestellt.

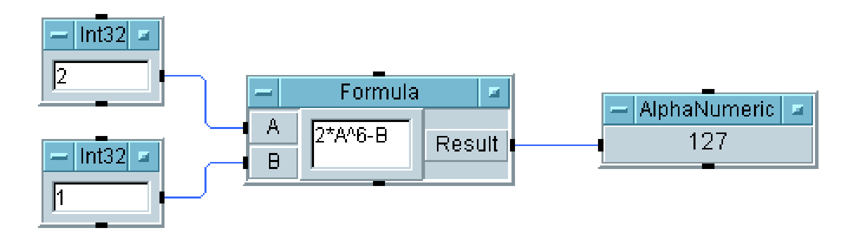


Abbildung 4-6. Auswerten eines Ausdrucks

Verwenden einer Agilent VEE-Funktion im Formelobjekt

Dieses Beispiel generiert eine Kosinuskurve und berechnet die Standardabweichung und den quadratischen Mittelwert mithilfe des `Formula`-Objekts.

1. Wählen Sie die Objekte `Function Generator`, `Formula` und `AlphaNumeric` aus und verbinden Sie sie mit ihren Daten-Pins.

2. Klonen Sie das `Formula`-Objekt durch Öffnen des Objektmenüs und Auswählen von `Clone`. Platzieren Sie dieses Objekt direkt unter dem ersten. Verbinden Sie den Datenausgangs-Pin des `Function Generator` mit dem zweiten `Formula`-Objekt.

3. Klonen Sie eine weitere AlphaNumeric-Anzeige und verbinden Sie sie mit dem zweiten `Formula`-Objekt.

4. Geben Sie `sdev(A)` im ersten `Formula`-Objekt und `rms (A)` im zweiten `Formula`-Objekt ein.

   `sdev(A)` und `rms (A)` sind die beiden mathematischen Funktionen aus dem Dialogfenster `Device ⇒ Function & Object Browser`. Beachten Sie, dass sie als Funktionen oder unabhängige Objekte aufgerufen werden können; sie werden in beiden Fällen auf die gleiche Weise ausgeführt.

5. Führen Sie das Programm aus. Das Programm zeigt die gleichen Antworten an, ob diese Funktionen im `Formula`-Objekt oder als unabhängige Objekte verwendet werden (siehe Abbildung 4-7).

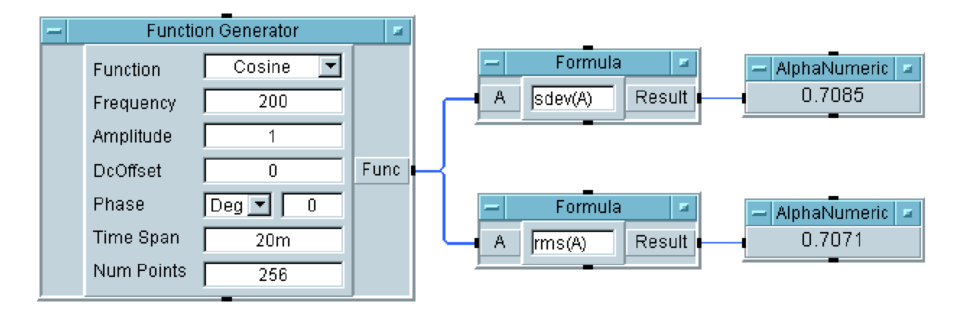


Abbildung 4-7. Formelbeispiele mit VEE-Funktionen

Berechnen Sie jetzt die Standardabweichung und den quadratischen Mittelwert mit nur einem `Formula`-Objekt. Formeln können mehrere Ausgangsanschlüsse mit Werten zugeordnet werden.

6. Klicken Sie doppelt auf das Objektmenü, um eines der `Formula`-Objekte mit `Cut` auszuschneiden.

7. Ändern Sie in dem verbleibenden `Formula`-Objekt den Ausdruck in
   ```
   B=sdev(A);
   C=rms(A)
   ```

Hinweis

Wenn ein `Formula`-Objekt mehrere Ausdrücke enthält, müssen Sie ein Semikolon am Ende eines Ausdrucks einfügen, um den Ausdruck von dem nächsten abzugrenzen. In der Formel `B=sdev(A);` kennzeichnet das Semikolon beispielsweise das Ende des Ausdrucks.

Hinweis

Sie können *Zeilenumbrüche* an jeder beliebigen Stelle in einem `Formula`-Objekt einfügen. Die Formel wird als ein einziger Ausdruck gelesen, wenn keine Semikolons darin enthalten sind. Sie können beispielsweise einen einzelnen Ausdruck eingeben als

```
B=sdev
(A)
```

Sie können auch Leerzeichen in die Formel einfügen, um die Lesbarkeit zu verbessern.

8. Fügen Sie in dem `Formula`-Objekt einen Ausgangsanschluss hinzu. Benennen Sie die Ausgangsanschlüsse um in `B` und `C`. Verbinden Sie den Ausgangsanschluss `B` mit einem der `Alphanumeric`-Objekte und den Ausgangsanschluss `C` mit dem anderen `Alphanumeric`-Objekt.

9. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 4-8.

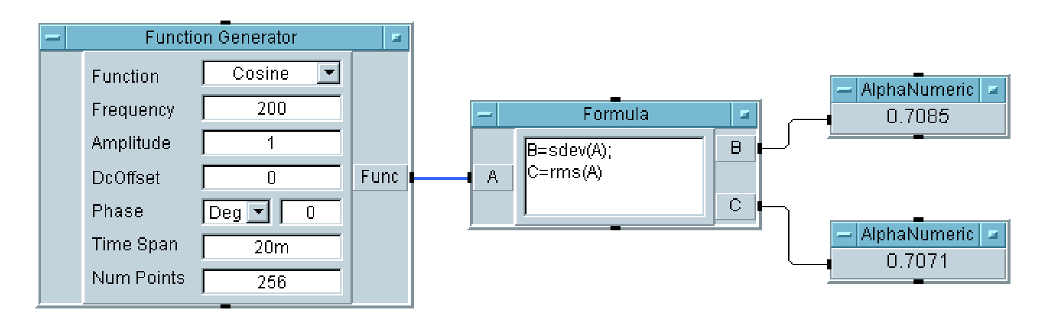


Abbildung 4-8. VEE Funktionen mit einem Formelobjekt

Eigene Experimente

Führen Sie die folgende Übung durch und überprüfen Sie die Ergebnisse (siehe Abbildung 4-9).

1. Erstellen Sie einen Array von Zahlen von 1 bis 2048 mit dem Objekt `ramp` in der Kategorie `Generate` der `Built-in Functions`. Berechnen Sie die Standardabweichung dieses Arrays und zeigen Sie sie an.

2. Führen Sie die gleiche Übung durch wie im vorigen Schritt; verwenden Sie dabei die Funktion `ramp()` in einem `Formula`-Objekt statt des Objekts `ramp`.

3. Führen Sie die im vorigen Schritt beschriebene Übung durch, indem Sie die Funktionen verschachteln. Verwenden Sie nur zwei Objekte.

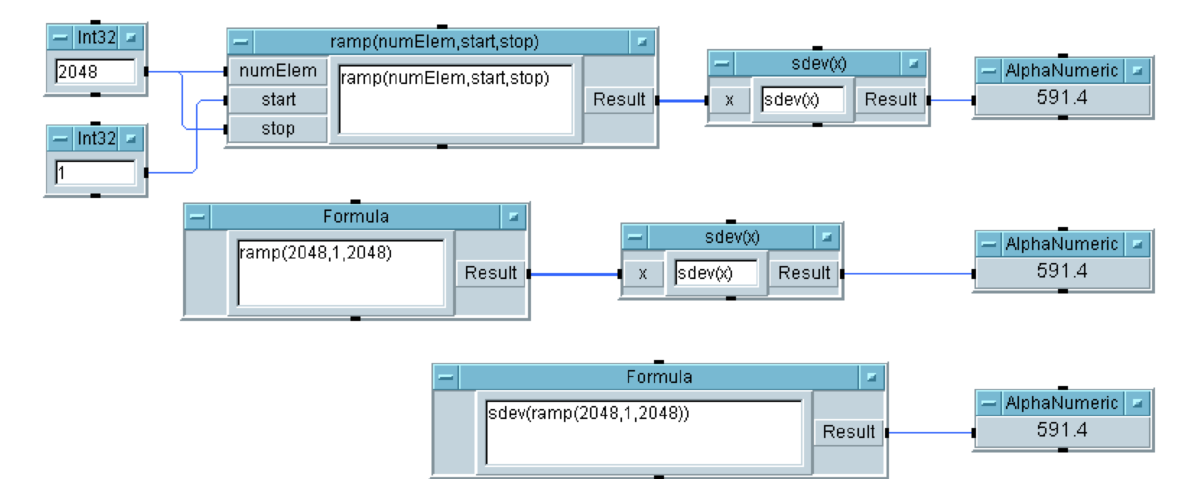


Abbildung 4-9. Lösung zu den eigenen Experimenten: Ramp und SDEV

Für die zweite und dritte Übung müssen Sie den Eingangsanschluss `A` am `Formula`-Objekt löschen, um eine Fehlermeldung zu vermeiden, da alle Dateneingangs-Pins angeschlossen sein und Daten aufweisen müssen, bevor ein Objekt ausgeführt werden kann.

Verwenden von MATLAB Script in Agilent VEE

VEE umfasst das Objekt "MATLAB Script", mit dem Sie Zugriff auf die Funktionalität von MATLAB haben. VEE kann Daten an die MATLAB Script-Engine übergeben und umgekehrt Daten von dieser Engine empfangen. Somit können Sie mathematische MATLAB-Funktionen in VEE-Programme einbeziehen.

Hinweis

Wenn Sie MATLAB bereits installiert haben, verwendet VEE das installierte MATLAB-Produkt zur Verarbeitung des MATLAB Script. Falls Sie die Signalverarbeitungs-Toolbox nicht haben, können Sie diese Funktionen von VEE aus nicht nutzen, es sei denn, die mit VEE ausgelieferte MATLAB Script Engine wurde registriert. Zum Registrieren von MATLAB wechseln (CD) Sie zum Verzeichnis <*VEE_installation_dir*>`\MATLAB\bin` und starten das Programm
`MATLAB.exe /regserver`.

Einige der Anwendungen für das MATLAB Script-Objekt sind:

- MATLAB-Verarbeitung von Daten, die mit VEE generiert wurden.
- Zurückgeben von Ergebnissen vom MATLAB Script-Objekt und Verwenden dieser Ergebnisse in anderen Teilen des VEE-Programms.
- Ausführen von komplexen Filter-Designs und Implementierungen im MATLAB Script-Objekt durch Verwenden der Funktionalität der MATLAB Signalverarbeitungs-Toolbox.
- Darstellung von Daten als 2-D- oder 3D-Grafiken.

Abbildung 4-10 zeigt, wie das MATLAB Script-Objekt in einem VEE-Programm erscheint. Bei der Ausführung des MATLAB Script-Programms generiert es die im `Alphanumeric`-Objekt angezeigten Daten.

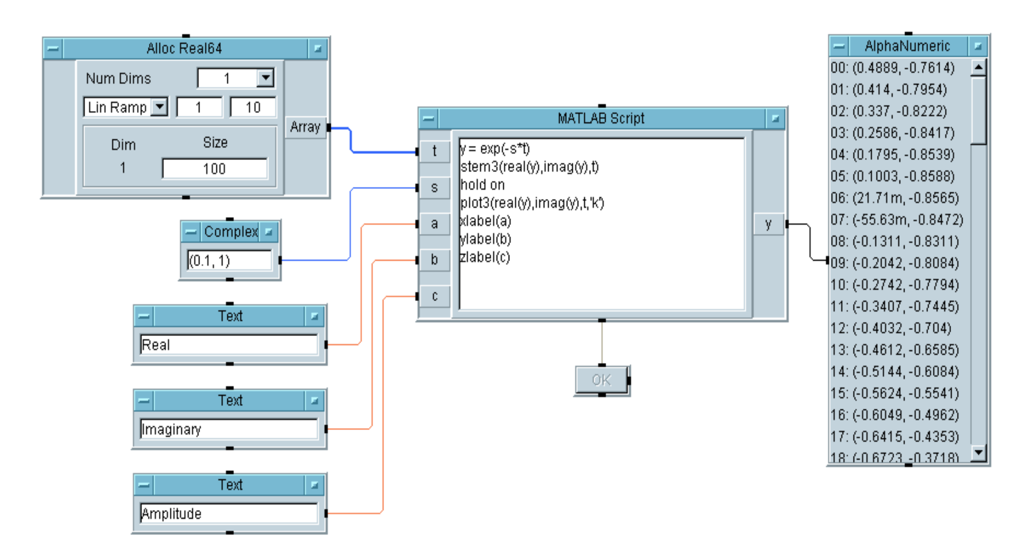


Abbildung 4-10. MATLAB Script-Objekt in einem VEE-Programm

Abbildung 4-11 zeigt die bei der Ausführung des Programms generierte Grafik an.

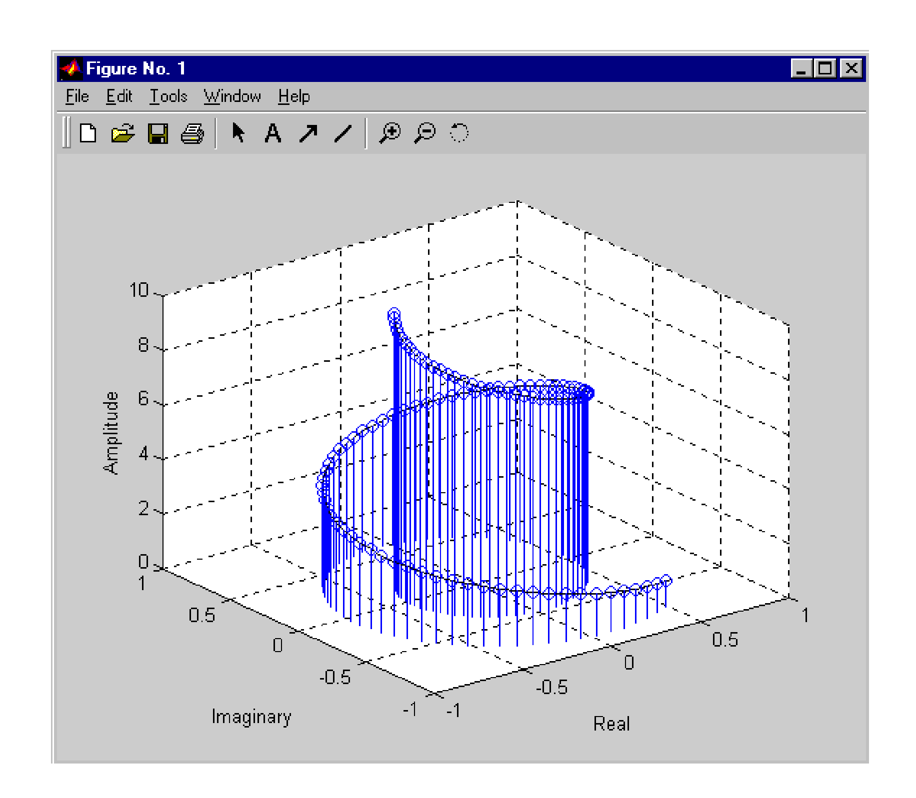


Abbildung 4-11. Vom Programm generierte Grafik

Wenn Sie MATLAB Script-Objekte in einem VEE-Programm verwenden, ruft VEE die MATLAB Script-Engine auf zur Ausführung von Operationen in den MATLAB Script-Objekten. Informationen werden von VEE an MATLAB übergeben und umgekehrt. Einige Hinweise zu MATLAB:

- Das erste in einem Programm ausgeführte MATLAB Script-Objekt öffnet eine einzelne MATLAB-Session. Alle weiteren Exemplare von MATLAB Script-Objekte nutzen diese Session gemeinsam. MATLAB Script-Objekte können daher globale Variablen im MATLAB-Arbeitsbereich gemeinsam nutzen.

- VEE führt *keine* Syntaxprüfung der MATLAB-Befehle vor dem Aufruf der MATLAB Script-Engine durch. Von MATLAB generierte Fehler und Warnungen werden in den normalen VEE-Dialogfenstern angezeigt wie alle anderen VEE-Fehler- und Warnungsmeldungen.
- Im Gegensatz zu VEE wird bei MATLAB zwischen Groß- und Kleinbuchstaben unterschieden. Wenn Sie einen Ein- oder Ausgangsanschluss eines MATLAB Script-Objekts mit dem Großbuchstaben `X` benennen, vergewissern Sie sich, dass Sie in MATLAB wirklich den Großbuchstaben `X` verwenden und nicht etwa den Kleinbuchstaben `x`.
- Als Eingang für MATLAB Scripts sind nur bestimmte VEE-Datentypen zulässig. Diese Typen werden im nächsten Abschnitt beschrieben.

Einbeziehen eines MATLAB Script-Objekts in Agilent VEE

Wenn Sie ein MATLAB-Objekt in ein VEE-Programm einbeziehen, sieht es aus wie ein VEE-`Formula`-Objekt. Es gibt zwei Möglichkeiten, ein MATLAB Script-Objekt einem Programm hinzuzufügen:

1. Wählen Sie `Device` ⇒ `MATLAB Script` aus und klicken Sie mit der Maustaste, um das Objekt im Programm zu platzieren. Dadurch wird ein Standard MATLAB Script-Objekt erstellt, das Sie für Ihre Zwecke anpassen können.

   -ODER-

   Wählen Sie Device ⇒ `Function & Object Browser` und anschließend `Type: MATLAB Functions` aus. Wählen Sie eine vordefinierte MATLAB-Function aus und klicken Sie `Create Formula` an. Klicken Sie mit der Maustaste, um das Objekt im Programm zu platzieren. Abbildung 4-12 zeigt einige vordefinierte MATLAB-Funktionen, die Sie einem VEE-Programm hinzufügen können.

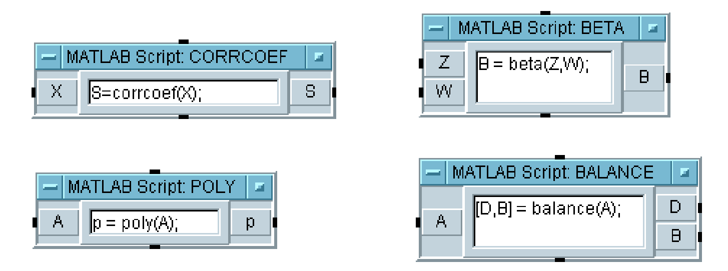


Abbildung 4-12. Hinzufügen vordefinierter MATLAB-Objekte zu einem VEE-Programm

Beachten Sie, dass jedes Objekt den Namen `MATLAB Script`<*function name*> erhält, damit Sie es von anderen VEE-Formelobjekten unterscheiden können. Jedes Objekt enthält bereits die Funktion, die es ausführen wird, sowie die voraussichtlich erforderlichen Eingangs- und Ausgangs-Pins, genau wie bei integrierten VEE-Formelobjekten. Sie können MATLAB Script-Objekte auch bearbeiten, genau wie andere VEE-Objekte.

Hinweis

Weitere Informationen zu MATLAB-Funktionen können Sie über das VEE-Hauptfenster durch Auswahl von `Help` ⇒ `MATLAB Script` ⇒ `Help Desk` aufrufen.

Arbeiten mit Datentypen

Nur eine Untermenge der VEE-Datentypen werden als Ein- und Ausgang von MATLAB-Objekten unterstützt.

VEE wandelt manche eindimensionalen Arrays automatisch um, um die Verarbeitung durch Programme zu ermöglichen, die sowohl VEE- als auch MATLAB-Funktionen enthalten. Ein eindimensionaler VEE-Text-Array wird z. B. automatisch in einen zweidimensionalen Zeichen-Array umgewandelt, wenn er als Eingang für ein MATLAB Script-Objekt verwendet wird; ein eindimensionaler Zeichen-Array aus einem MATLAB Script-Objekt wird automatisch in ein Text-Skalar umgewandelt, wenn er den Ausgang eines MATLAB Script-Objekts darstellt.

Hinweis Eine vollständige Liste und Beschreibung über die automatischen Umwandlungen zwischen VEE- und MATLAB-Datentypen finden Sie in der VEE-Online-Hilfe.

Sie können auch Begrenzungen für die Datentypen am Eingangsanschluss verwenden, um sicherzustellen, dass der Dateneingang von einem anderen Objekt in einen unterstützten Typ umgewandelt wird, wie im folgenden Beispiel gezeigt.

1. Wählen Sie `Data` ⇒ `Constant` ⇒ `Int32` aus und klicken Sie mit der Maustaste, um das Objekt zu platzieren. Ändern Sie den Wert in 7. Klonen Sie das Objekt mit `Clone`, und platzieren Sie das zweite `Int32`-Objekt unter dem ersten. Ändern Sie seinen Wert in 20.

2. Wählen Sie `Device` ⇒ `MATLAB Script` aus und platzieren Sie das Objekt rechts von den Konstantenobjekten.

3. Wählen Sie `Display Alphanumeric` aus und platzieren Sie das Objekt rechts von dem `MATLAB Script`-Objekt.

4. Verbinden Sie den Ausgangs-Pin des oberen `Int32`-Objekts mit dem Eingangs-Pin `A` des `MATLAB Script`-Objekts. Verbinden Sie den Ausgangs-Pin des unteren `Int32`-Objekts mit dem Eingangs-Pin `B` des `MATLAB Script`-Objekts. Verbinden Sie den Ausgangs-Pin des `MATLAB Script`-Objekts mit dem Eingangs-Pin des `Alphanumeric`-Objekts.

   Führen Sie das Programm aus. Es generiert einen VEE-Laufzeitfehler mit der Angabe, dass als Eingang `Real64`, `Complex`, `Waveform` oder `Text` erwartet, aber `Int32` als Eingang empfangen wurde.

Sie können Fehler wie diesen vermeiden, indem Sie den Datentyp am Eingangsanschluss des MATLAB Script-Objekts ändern.

5. Klicken Sie doppelt auf den Anschluss `A`, um das Dialogfenster `Input Terminal Information` zu öffnen. Klicken Sie `Required Type:` an, um das Dropdown-Menü anzuzeigen, wählen Sie `Real64` aus, und klicken Sie `OK` an. Klicken Sie doppelt auf den Anschluss `B` und ändern Sie ihn ebenfalls in `Real64`, wie in Abbildung 4-13 gezeigt.

6. Führen Sie das Programm aus. Die `Int32`-Daten werden jetzt am Eingangs-Pin automatisch in `Real64` umgewandelt, bevor sie an MATLAB weitergegeben werden.

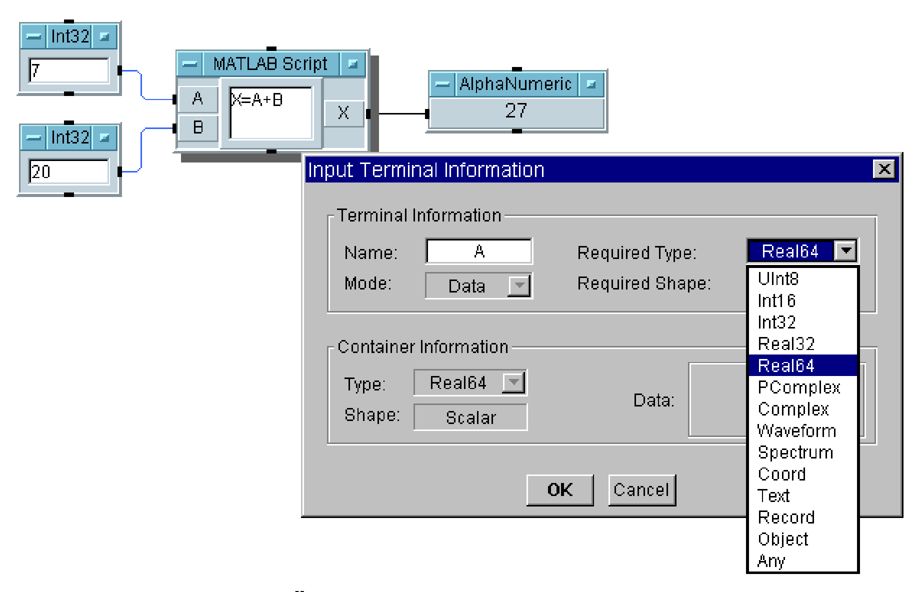


Abbildung 4-13. Ändern des Datentyps am Eingangsanschluss

Anzeigen von Testdaten

Tabelle 4-2 beschreibt die Anzeigefunktionen für die verschiedenen VEE-Objekte.

Tabelle 4-2. Anzeigen

Anzeige	Beschreibung
Alphanumeric	Anzeigen von Werten als Text oder als Zahlen. Erfordert SCALAR, ARRAY 1D oder ARRAY 2D.
Beep	Gibt einen hörbaren Ton aus, um eine Stelle in Ihrem Programm zu kennzeichnen.
Complex Plane	Zeigt die Datenwerte Complex, Polar Complex (PComplex) oder Coord an einer realen gg. einer imaginären Achse an.
Indicator=>> Meter, Thermometer, Fill Bar, Tank, Color Alarm	Alle diese Indikatoren zeigen Zahlen mit einer über ihre Namen naheliegenden grafischen Darstellung an. Alle diese Indikatoren haben farbig codierte Bereiche - normalerweise drei, "Meter" sogar fünf. Der "Color Alarm" kann eine bei einem Alarm in den verschiedenen Bereichen blinkende LED über eine blinkende Textmeldung simulieren.
Label	Ein Objekt, mit dem eine Textbeschriftung in der Fensteransicht platziert wird. Die Farben und Schriften können in der Fensteransicht über "Properties..." im Objektmenü leicht angepasst werden.
Logging Alphanumeric	Zeigt bei einer wiederholten Aufzeichnung Werte als Text oder Zahlen an. Erfordert SCALAR oder ARRAY 1D.
Note Pad	Verwendet eine Textnotiz zur Verdeutlichung eines Programms.

Tabelle 4-2. Anzeigen (Fortsetzung)

Anzeige	Beschreibung
Picture (PC)	Ein Objekt, mit dem eine grafische Beschriftung in der Fensteransicht platziert wird. Die unterstützten Formate sind: *.BMP (Bitmaps), *.GIF (GIF87a und GIF89), *.ICN (X11-Bitmap), *.JPEG, *.PNG und *.WMF (Windows Meta File)
Picture (UNIX)	*.GIF (GIF87a) und *.xwd (X11 Window Dump)
Polar Plot	Zeigt Daten grafisch auf einer polaren Skala an, wenn separate Informationen für Radius- und Winkeldaten verfügbar sind.
Spectrum (Freq)	Ein Menü, das Frequenzbereichanzeigen enthält: Magnitude Spectrum, Phase Spectrum, Magnitude gg. Phase (Polar) und Magnitude gg. Phase (Smith). Der Eingang muss Waveform, Spectrum oder ein Koordinaten-Array sein. Waveform-Eingänge werden über eine a Fast Fourier-Umwandlung (fft) automatisch in Frequenzbereiche umgewandelt.
Strip Chart	Zeigt die letzte Historie der bei der Ausführung des Programms laufend generierten Daten grafisch an. Für jeden y-Eingangswert wird der x-Wert um eine angegebene Schrittgröße hochgezählt. Wenn neue Daten über die rechte Seite der Anzeige hinausragen, blättert die Anzeige automatisch nach rechts, um die neuesten Daten anzuzeigen.
Waveform (Time)	Zeigt Wellenformen (Waveform) oder Spectrums im Echtzeitbereich grafisch an. Spectrums werden über eine invertierte Fast Fourier-Umwandlung (ifft) automatisch in den Zeitbereich umgewandelt. Die x-Achse entspricht den Sampling-Einheiten der Eingangs-Wellenform.
X gg. Y Plot	Zeigt Werte grafisch an, wenn separate Dateninformationen für X- und Y-Daten verfügbar sind.

Tabelle 4-2. Anzeigen (Fortsetzung)

Anzeige	Beschreibung
XY Trace	Zeigt zugeordnete Arrays oder eine Gruppe von Werten grafisch an, wenn y-Daten mit gleichmäßig verteilten x-Werten generiert werden. Der automatisch generierte x-Wert hängt von Datentyp der Trace-Daten ab. Ein Real-Trace generiert beispielsweise gleichmäßig verteilte reale x-Werte; ein Waveform-Trace generiert dagegen x-Werte für die Zeitachse.

Anpassen von Testdatenanzeigen

Die Anzeigen können auf verschiedene Arten angepasst werden. Sie können die Anzeigen nicht nur beschriften, verschieben und ihre Größe ändern wie bei anderen VEE-Objekten auch, sondern Sie können auch die x/y-Skala und die Traces ändern, Markierungen hinzufügen oder Teile der Anzeige in der grafischen Anzeigen zoomen.

Das folgende Beispiel verdeutlicht einige dieser Funktionen. Es verwendet den `Noise Generator` zum Generieren einer Wellenform und zeigt diese anschließend mit der Anzeige `Waveform (Time)` an. Das Beispiel zeigt auch, wie die `X`-Skala geändert, ein Segment der Wellenform gezoomt und die Markierungen zum Messen der Abstände zwischen Punkten der Wellenform verwendet werden können. Dasselbe Prinzip kann auf alle grafischen Anzeigen angewendet werden.

Anzeigen einer Wellenform

1. Wählen Sie `Device` ⇒ `Virtual Source` ⇒ `Noise Generator` aus.

2. Wählen Sie `Display` ⇒ `Waveform (Time)` aus.

3. Verbinden Sie den Datenausgang des `Noise Generator` mit dem Dateneingang von `Waveform (Time)` und führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 4-14.

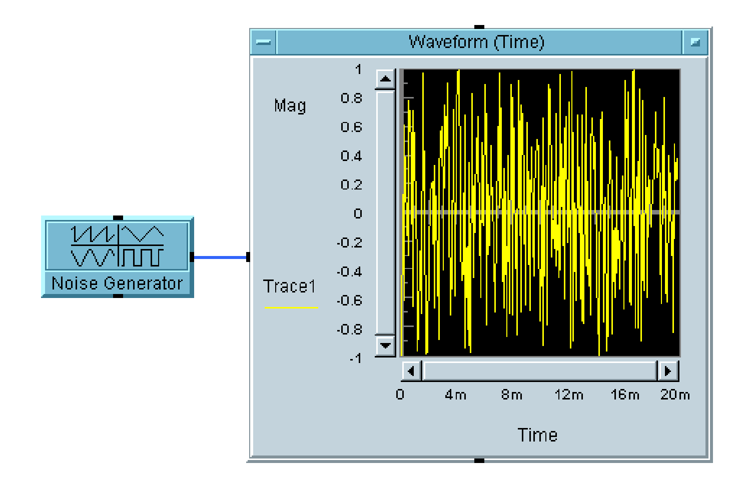


Abbildung 4-14. Anzeigen einer Wellenform

Ändern der X- und Y-Skala

1. Klicken Sie doppelt auf die `Waveform (Time)`-Titelleiste, um das Fenster `Y Plot Properties` zu öffnen, wählen Sie den Ordner `Scales` aus, wählen Sie `20m` als `X Maximum` aus und geben Sie `1m` ein.

   Dadurch wird das Zeitfenster der Anzeige von 20 Millisekunden auf 1 Millisekunde geändert.

2. Klicken Sie doppelt auf das Feld `Minimum` auf der `Y`-Achse, wo es `-1` angibt, und geben Sie den Wert `- .5` ein. Klicken Sie `OK` an.

Zoomen eines Teils der Wellenform

1. Öffnen Sie das `Waveform (Time)`-Objektmenü und klicken Sie `Zoom` ⇒ `In` an.

   Der Cursor wird als kleiner rechter Winkel angezeigt. Durch Klicken und Ziehen können Sie ein Quadrat in der Grafik zeichnen, das den zu zoomenden Bereich umschließt.

2. Ziehen Sie einen Umriss um einen Bereich der Wellenform, der mehrere Spitzen enthält, und lassen Sie die Maustaste los.

   Die Anzeige zoomt in den ausgewählten Bereich der Wellenform. Beachten Sie, dass die `x-` und `y`-Skala automatisch geändert wird.

Hinzufügen von Delta-Markierungen zur Anzeige

1. Wechseln Sie zur offenen Ansicht des `Noise Generator`.

   a. Ändern Sie die Einstellung von `Num Points` in `16`. Führen Sie das Programm erneut aus.

   b. Öffnen Sie das `Waveform (Time)`-Objektmenü und wählen Sie `Properties` aus (oder klicken Sie doppelt auf die Titelleiste). Klicken Sie anschließend unter `Markers` die Option `Delta` an. Klicken Sie `OK` an.

Hinweis Sie können die Werte der Markierungen zur Ausführungszeit abrufen und ändern. Weitere Informationen können Sie in der Online-Hilfe unter `Contents and Index` ⇒ `How Do I...` ⇒ `Display Data` anzeigen.

Sie sehen zwei weiße Pfeile an einem der Datenpunkte der Wellenform, die nach oben und unten weisen. Beachten Sie auch, dass die `x`- und `y`-Koordinaten dieser Markierungen am unteren Rand der Anzeige eingeblendet werden. Zum Messen des `x`- oder `y`-Abstands zwischen zwei Spitzen klicken-und-ziehen Sie die zu messenden Pfeile. Sie sehen, wie sich eine der Markierungen zu diesen neuen Spitzen verschiebt; die neuen Koordinaten werden am unteren Rand der Anzeige eingeblendet (siehe Abbildung 4-15).

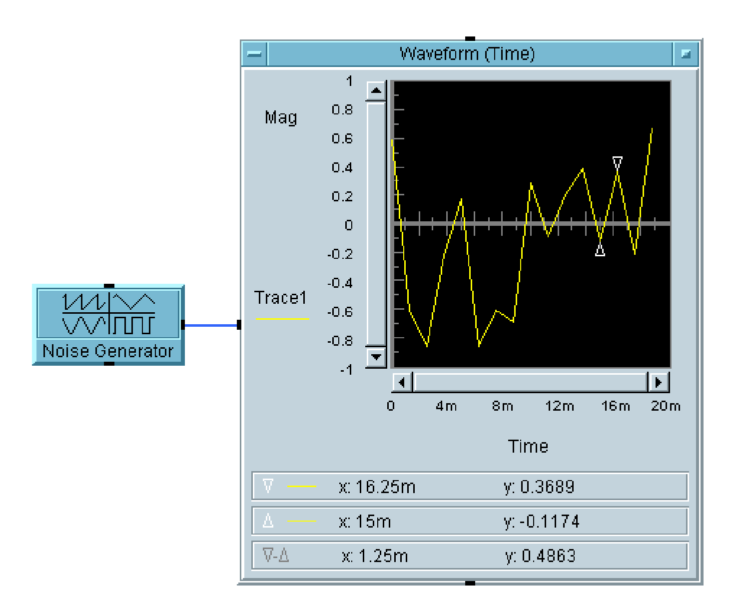


Abbildung 4-15. Delta-Markierungen in einer Wellenform-Anzeige

VEE interpoliert automatisch zwischen den Datenpunkten der Wellenform. Öffnen Sie das Objektmenü, wählen Sie `Properties` aus und klicken Sie unter `Markers` die Option `Interpolate` an.

Ändern der Farbe des Trace

1. Klicken Sie doppelt auf die Titelleiste, um das Fenster `Properties` aufzurufen, und klicken Sie anschließend den Ordner `Traces` an.

   Sie können die Farbe, den Linientyp und den Punkttyp für den ausgewählten Trace in diesem Ordner ändern.

Hinweis

Sie können diese Werte zur Ausführungszeit mit den Steuereingängen `Traces` oder `Scales` ändern. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

2. Klicken Sie `OK` für die ausgewählte Farbe an. Klicken Sie anschließend `OK` an, um das Fenster `Properties` zu schließen.

Der Trace wird jetzt in der neuen Farbe angezeigt. Weitere Anzeigemerkmale wie Fenster-Layout, Gittertyp, Löschkontrolle und das Hinzufügen der rechten Skala können auf ähnliche Weise angepasst werden wie die Funktionen in der Übung oben.

Hinweis

VEE enthält auch die Funktion `Plot` in den Anzeigen-Objektmenüs. Mit dieser Funktion können Sie die Testergebnisse auf der Anzeige plotten, ohne den Rest des Programms auszudrucken.

Zur weiteren Übung

Weitere Informationen zu anderen VEE-Objekten und zusätzliche Sicherheit beim Umgang mit Agilent VEE erhalten Sie, wenn Sie die Übungen in Anhang A, “Zusätzliche Übungen” auf Seite 493 durchführen. Die Lösungen mit Erläuterungen der wichtigen Punkte sind ebenfalls enthalten.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie zum nächsten Kapitel übergehen.

- Beschreiben der wichtigsten Datentypen in VEE.
- Beschreiben einiger der Hauptanwendungen für die Analysefunktionen in VEE.
- Anzeigen der Hilfe-Erläuterung zu jedem Objekt im Dialogfenster `Function & Browser`.
- Beschreiben der Beziehung zwischen Eingangs-Pins und Variablen in einem mathematischen VEE-Objekt.
- Auswerten eines mathematischen Ausdrucks mit dem `Formula`-Objekt und anschließendes Auswerten zweier Ausdrücke mit dem `Formula`-Objekt. (Denken Sie daran, nach der ersten Zeile ein Semikolon zu verwenden.)
- Verwenden einer VEE-Funktion in einem mathematischen Ausdruck im Formelobjekt.
- Verwenden des MATLAB Script-Objekts.
- Beschreiben der wichtigsten Anzeigefunktionen in VEE.
- Anpassen einer grafischen Anzeige in Bezug auf die verwendeten Skalen, den angezeigten Teil der Wellenform, die verwendeten Markierungen und die Farbe des Trace.

5

Speichern und Abrufen von Testergebnissen

Speichern und Abrufen von Testergebnissen

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Ablegen von Testdaten in Arrays
- Verwenden des Objekts `Collector`
- Verwenden des Objekts `To/From File`
- Erstellen gemischter Datentypen mit `Records`
- Ausführen von Such- und Sortieroperationen mit `DataSets`
- Erstellen einfacher Testdatenbanken mit den `Dataset`-Objekten

Erforderliche Zeit für dieses Kapitel: 2 Stunden

Überblick

In diesem Kapitel lernen Sie die Grundfunktionen zum Speichern und Abrufen von Testdaten kennen. Sie erstellen Arrays des richtigen Datentyps und der richtigen Größe für Ihre Testergebnisse und rufen diese Daten oder Teile davon zur Analyse oder Anzeige auf.

In diesem Kapitel werden außerdem die Objekte `To/From File`, der Datentyp "Record" (Datensatz) und die Dataset-Dateien beschrieben. Die Objekte `To File` und `From File` lesen über E/A-Transaktionen Daten in und aus Dateien ein. Der Datentyp `Record` kann zum Speichern verschiedener Arten von Daten in einer einzigen Struktur verwendet werden. Sie können den `Dataset` verwenden, um ein oder mehrere Datensätze in einer Datei zu speichern sowie Such- und Sortieroperationen mit Datasets ausführen.

Hinweis Das Objekt `To File` ist auch im Abschnitt “Verwenden von Datendateien” auf Seite 2-3 in Kapitel 2, “Agilent VEE Programmiertechniken” beschrieben.

Verwenden von Arrays zum Speichern von Testergebnissen

Datentypen können auf zwei Arten gespeichert werden:

- Als Skalarwerte (einzelne Zahlen wie beispielsweise 9 oder (32, @10))

-ODER-

- Als Arrays mit 1 bis 10 Dimensionen

Hinweis Eine Übersicht über die VEE-Datentypen finden Sie in Kapitel 4, "Analysieren und Anzeigen von Testdaten".

Die Indexierung für Arrays erfolgt in VEE mit der Basis Null, die Position des Array-Elements wird mit Klammern angegeben. Wenn der Array `A` beispielsweise die Elemente `[4, 5, 6]` enthält, gilt:

`A[0] = 4`, `A[1] = 5` und `A[2] = 6`

Die Syntax für Arrays lautet wie folgt:

Doppelpunkt	Kennzeichnet einen Bereich von Elementen, im obigen Beispiel-Array etwa `A[0:2]` = `[4, 5, 6]`.
Sternchen (*)	Ein Platzhalterzeichen zur Angabe aller Elemente einer bestimmten Array-Dimension. `A[*]` gibt alle Elemente eines Array `A` zurück.
Kommas	In der Syntax von Unter-Arrays werden die Array-Dimensionen mit Kommas getrennt. Wenn `B` ein zweidimensionaler Array mit drei Elementen in jeder Dimension ist, gibt `B[1,0]` das erste Element in der zweiten Reihe von `B` zurück.

Die Syntax für den Zugriff auf Elemente eines Array kann im `Formula`-Objekt oder einem Ausdrucksfeld, beispielsweise im Objekt `To/From File`, verwendet werden.

Übung 5-1: Erstellen eines Array für Testergebnisse

Die einfachste Möglichkeit zum Erstellen eines Array ist die Verwendung des Objekts `Collector`.

In dieser Übung wird das Objekt `For Count` zum Simulieren von vier Einlesevorgängen von einem Instrument verwendet. Die "eingelesenen" Daten werden in einem Array abgelegt, und die Ergebnisse werden gedruckt. Das Prinzip ist unabhängig vom Datentyp oder der Größe des Array immer gleich, da der `Collector` beliebige Datentypen verwendet und die Größe des Array entsprechend der Anzahl der gesendeten Elemente erstellt.

1. Wählen Sie `Flow ⇒ Repeat ⇒ For Count`, `Data ⇒ Collector` und `Display ⇒ AlphaNumeric` aus.

Informationen zum Objekt For Count	`For Count` gibt aufsteigende ganzzahlige Werte aus, beginnend mit 0, entsprechend der im Eingabefeld angegebenen Iterationen. Heben Sie die Standardanzahl `10` durch Doppelklicken hervor, und geben Sie anschließend `4` ein. `For Count` gibt `0`, `1`, `2` und `3` aus.
Informationen zum Objekt Collector	Der `Collector` empfängt Datenwerte über seinen Dateneingangsanschluss. Wenn Sie mit dem Erfassen von Daten fertig sind, führen Sie einen “ping” auf den Anschluss `XEQ` aus, um dem `Collector` mitzuteilen, dass der Array erstellt und ausgegeben werden soll. Sie können den Sequenzausgangs-Pin des Objekts `For Count` verwenden, um den "ping" auf den `Collector XEQ` auszuführen. Der `Collector` zeigt eine Schaltfläche an zum Umschalten zwischen einem `1 Dim Array` und einem *n+1* `Dim Array`. Klicken Sie doppelt auf den `Collector`, um die offene Ansicht aufzurufen, und lesen Sie unter `Help` im Objektmenü nach, um die Arbeitsweise des Objekts zu verstehen.

2. Klicken Sie `n+1 Dim` im `Collector` an, um die Auswahl in `1 Dim Array` zu ändern.

3. Verbinden Sie den Datenausgangs-Pin des Objekts `For Count` mit dem Dateneingangs-Pin am `Collector`.

4. Verbinden Sie den Sequenzausgangs-Pin des Objekts `For Count` mit dem Eingangs-Pin `XEQ` am `Collector`.

   Der XEQ-Pin, ein spezieller Auslöser-Pin, den es bei verschiedenen Objekten gibt, legt fest, wann das Objekt ausgeführt wird. In diesem Fall soll das Objekt ausgelöst werden, wenn alle Daten für den Array erfasst wurden.

5. Verbinden Sie den Datenausgangs-Pin des `Collector` mit dem Dateneingangs-Pin des `AlphaNumeric`-Objekts.

6. Vergrößern Sie `AlphaNumeric`, um genügend Platz für den Array zu schaffen. Klicken und ziehen Sie hierzu eine der Ecken des Objekts. (Sie könnten `AlphaNumeric` auch schon beim ersten Aufruf vergrößern, indem Sie seine Umrisslinie anklicken und ziehen.)

7. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 5-1.

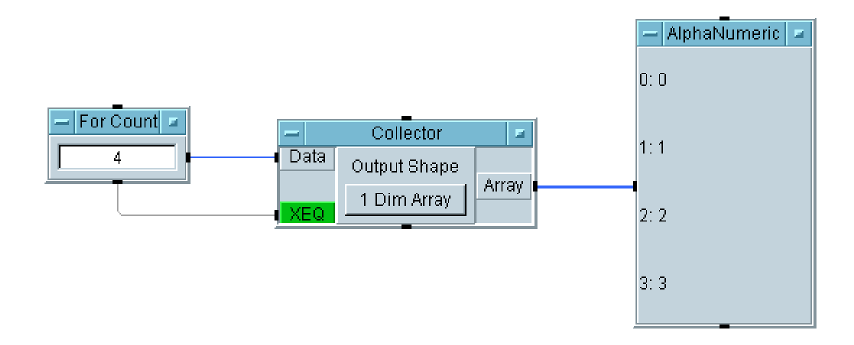


Abbildung 5-1. Der Collector erstellt einen Array

Übung 5-2: Extrahieren von Werten aus einem Array

Zum Extrahieren von Werten aus einem Array können Sie die Klammersyntax in einem Ausdruck oder das Objekt `Access Array ⇒ Get Values` verwenden. Im folgenden Beispiel werden Ausdrücke im `Formula`-Objekt verwendet. Sie fügen in dieser Übung dem Programm mehrere Objekte hinzu.

1. Löschen Sie die Datenlinie zwischen dem `Collector` und dem `AlphaNumeric`-Objekt, indem Sie den Mauszeiger auf der Linie positionieren, **Umschalttaste-Strg** drücken und mit der linken Maustaste klicken. Stellen Sie anschließend den `Collector` als Symbol dar.

2. Wählen Sie `Device ⇒ Formula` aus und klonen Sie das Objekt. Verschieben Sie `AlphaNumeric` nach rechts, und platzieren Sie beide `Formula`-Objekte rechts vom `Collector`.

3. Verbinden Sie den Datenausgang des `Collector` mit dem Dateneingang der `Formula`-Objekte. Geben Sie `A[2]` im oberen Formeleingabefeld ein und `A[1:3]` im unteren Formeleingabefeld.

   `A[2]` extrahiert das dritte Element des Array als Skalar; `A[1:3]` gibt einen Unter-Array aus drei Elementen mit dem zweiten, dritten und vierten Element von `A` (der Array des `A`-Eingangsanschlusses) zurück.

4. Klonen Sie `AlphaNumeric` und verbinden Sie eine Anzeige mit jedem `Formula`-Objekt.

5. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 5-2.

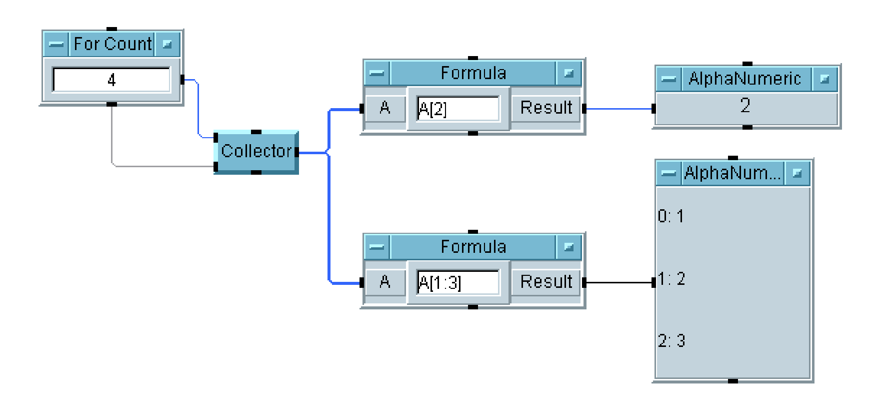


Abbildung 5-2. Extrahieren von Array-Elementen mit Ausdrücken

Verwenden der Objekte To/From File

Die Objekte `To File` und `From File` lesen über E/A-Transaktionen Daten in und aus Dateien ein. Sie haben folgende Merkmale:

- Bei der ersten `READ`- oder `WRITE`-Transaktion wird eine Datendatei geöffnet. Beim Beenden des Programms schließt VEE alle geöffneten Dateien automatisch.

- VEE verwaltet einen Lesezeiger und einen Schreibzeiger pro Datei, unabhängig davon, wie viele Objekte auf die Datei zugreifen. Der Lesezeiger kennzeichnet die Daten, die als nächste gelesen werden, und der Schreibzeiger kennzeichnet die Stelle, an der das nächste Datenelement geschrieben wird.

- Die Objekte `To/From File` können Daten an vorhandene Dateien anhängen oder die Daten überschreiben. Wenn die Einstellung `Clear File at PreRun & Open` in der offenen Ansicht des Objekts `To File` markiert ist, verweist der Schreibzeiger auf den Anfang der Datei. Ist diese Option nicht markiert, verweist der Zeiger auf das Ende der vorhandenen Datei. Jede `WRITE`-Transaktion hängt Daten an der Position des Schreibzeigers an die Datei an. Beim Ausführen einer `EXECUTE CLEAR`-Transaktion wird der Schreibzeiger an den Anfang der Daten verschoben und löscht den Inhalt der Datei.

- Ein Lesezeiger beginnt am Anfang der Datei und wandert entsprechend den `READ`-Transaktionen vorwärts durch die Datei. Sie können im Objekt `From File` eine `EXECUTE REWIND`-Transaktion ausführen, um den Zeiger wieder an den Anfang der Datei zu verschieben; dies hat keine Auswirkung auf die Daten.

Hinweis

Das Objekt `To File` ist auch im Abschnitt “Verwenden von Datendateien” auf Seite 2-3 in Kapitel 2, “Agilent VEE Programmiertechniken” beschrieben.

Die Arbeitsweise der E/A-Transaktionen

E/A-Transaktionen werden von VEE zur Kommunikation mit Instrumenten, Dateien, Zeichenfolgen, dem Betriebssystem, Schnittstellen, anderen Programmen, Rocky Mountain Basic und Druckern verwendet. Sehen Sie sich als Beispiel das Objekt `To File` in Abbildung 5-3 an.

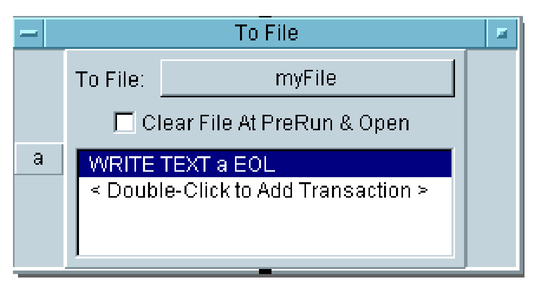


Abbildung 5-3. Das Objekt "To File"

Das in Abbildung 5-3 dargestellte Objekt `To File` sendet Daten an das angegebene Objekt `myFile`. Es kann Eingänge, die so genannten Transaktionen, enthalten, die Daten von einem Programm empfangen. Dieses Objekt `To File` enthält beispielsweise die Transaktion `WRITE TEXT a EOL`. Wenn Sie die Transaktion doppelt anklicken, erscheint ein Dialogfenster `I/O Transaction` wie in Abbildung 5-4 dargestellt; über dieses Dialogfenster wird die spezifische Transaktionsanweisung konfiguriert.

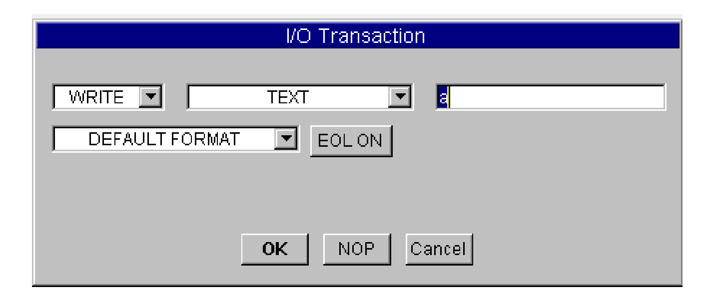


Abbildung 5-4. Ein Dialogfenster "I/O Transaction"

Je nach dem Objekt gibt es verschiedene Formen dieses Dialogfensters, alle diese Dialogfenster enthalten jedoch einige gemeinsame Elemente wie die “Aktionen", die "Codierung", die "Ausdrucksliste", das "Format" und die "Zeilenende-Sequenz" (EOL).

E/A-Transaktionsformat

Eine E/A-Transaktion zum Schreiben von Daten hat normalerweise das folgende Format:

<Aktion> <Codierung> <Ausdrucksliste> <Format> <EOL>

Tabelle 5-1 beschreibt die gängigsten Aktionen: READ, WRITE, EXECUTE und WAIT.

Tabelle 5-1. Typen von E/A-Transaktionen

Aktion	Erläuterung
READ	Liest Daten von der angegebenen Quelle mit der angegebenen Codierung und dem angegebenen Format ein.
WRITE	Schreibt Daten an das angegebene Ziel mit der angegebenen Codierung und dem angegebenen Format.
EXECUTE	Führt einen angegebenen Befehl aus. EXECUTE REWIND positioniert beispielsweise einen Schreib- oder Lesezeiger einer Datei auf den Anfang der Datei, ohne den Inhalt zu löschen. EXECUTE CLOSE schließt eine geöffnete Datei.
WAIT	Wartet die angegebene Anzahl von Sekunden vor der Ausführung der nächsten Transaktion.

Hinweis

Sie können auch eine Reihe weiterer Aktionen für `I/O ⇒ Advanced I/O Operations` untersuchen. Sehen Sie sich hierzu die Objekte in dem Menü an.

Codierungen und Formate beziehen sich auf die Art, in der Daten verpackt und gesendet werden. Eine TEXT-Codierung sendet Daten beispielsweise als ASCII-Zeichen. Die TEXT-Codierung kann auf unterschiedliche Arten formatiert werden. Wenn Sie beispielsweise eine Folge von Buchstaben und Ziffern an eine Datei senden wollen, wird diese Folge mit einer `WRITE TEXT STRING`-Transaktion als ASCII-Zeichen gesendet. Eine `WRITE TEXT REAL`-Transaktion dagegen extrahiert nur die Realen Zahlen aus der Zeichenfolge und sendet ASCII-Zeichen für die einzelnen Stellen der Zahlen. Tabelle 5-2 bietet eine kurze Erläuterung der Codierungen.

Tabelle 5-2. E/A-Transaktionscodierung

Codierung	Erläuterung
TEXT	Liest oder schreibt alle Datentypen in einer für den Benutzer lesbaren Form (ASCII); diese Form kann leicht bearbeitet oder in andere Software-Anwendungen portiert werden. Numerische VEE-Daten werden automatisch in Text umgewandelt.
BYTE	Wandelt numerische Daten in binäre Ganzzahlen um und sendet oder empfängt das letzte signifikante Byte.
CASE	Ordnet einen Numerierungswert oder eine Ganzzahl einer Zeichenfolge zu und liest/schreibt diese Zeichenfolge. Sie können CASE beispielsweise verwenden, um Fehlernummern zu empfangen und Fehlermeldungen zu schreiben.
BINARY	Verarbeitet alle Datentypen in einem maschinenspezifischen Binärformat.
BINBLOCK	Verwendet IEEE488.2 Block-Headers mit definierter Länge mit allen VEE-Datentypen in binären Dateien.
CONTAINER	Verwendet ein VEE-spezifisches Textformat mit allen Datentypen.

In einer WRITE-Transaktion ist eine "Ausdrucksliste" einfach eine durch Kommas begrenzte Liste von Ausdrücken, die ausgewertet werden müssen, um die gesendeten Daten zu liefern. Der Ausdruck kann aus einem mathematischen Ausdruck, dem Namen eines Dateneingangsanschlusses, einer Zeichenfolgekonstanten, einer VEE-Funktion, einer `UserFunction` oder einer globalen Variablen bestehen. In einer READ-Transaktion muss die Ausdrucksliste aus einer durch Kommas begrenzten Liste der Namen von Ausgangsanschlüssen bestehen, die angeben, wo die Daten nach dem Einlesen gespeichert werden sollen.

Die Datenformate werden in Verbindung mit dem Lesen von Daten aus Instrumenten in Kapitel 3, “Einfache Methoden zum Steuern von Instrumenten” auf Seite 133 beschrieben. Die meisten dieser Formate gelten für alle E/A-Transaktionen.

EOL (Zeilenende-Zeichenfolge) kann ein- oder ausgeschaltet werden. Sie können die EOL-Sequenz definieren, indem Sie das Objektmenü der meisten Objekte `I/O` ⇒ `To` öffnen und `Properties...` und anschließend `Data Format` auswählen, und die gewünschten Änderungen unter `Separator Sequence` vornehmen.

Übung 5-3: Verwenden der Objekte "To/From File"

In dieser Übung wird beschrieben, wie Testdaten in Dateien geschrieben und daraus gelesen werden. Sie werden in dieser Übung drei allgemeine Testergebnisse speichern und abrufen: einen Testnamen, einen Zeitstempel und einen eindimensionalen Array aus Real-Werten. Der gleiche Prozess gilt für alle VEE-Datentypen.

Senden einer Textzeichenfolge an eine Datei

1. Wählen Sie `I/O` ⇒ `To` ⇒ `File` aus. Legen Sie die Einträge wie folgt fest:

filename	Verwenden Sie die Standarddatei `myFile`. Se können die Standarddatei ändern, indem Sie das Eingabefeld `To File` anklicken, um eine Liste der Dateien im Heimverzeichnis anzuzeigen.
Clear File At PreRun & Open	Markieren Sie dieses Feld. Standardmäßig hängt VEE neue Daten an das Ende einer vorhandenen Datei an. Wenn dieses Feld markiert ist, wird der Inhalt der Datei vor dem Schreiben der neuen Daten gelöscht.

2. Klicken Sie doppelt im Transaktionsbereich, um das Dialogfenster `I/O Transaction` aufzurufen. (Schlagen Sie ggf. unter Abbildung 5-3 und Abbildung 5-4 nach.)

   `WRITE TEXT a EOL` ist die Standardtransaktion. Sie schreibt Daten an einen Pin mit einer TEXT-Codierung und einer angegebenen EOL-Sequenz. Bei VEE wird nicht zwischen Groß- und Kleinbuchstaben unterschieden. Sie können für die Namen der Dateneingangs- und Datenausgangsanschlüsse Klein- oder Großbuchstaben verwenden.

Legen Sie die Einträge wie folgt fest:

a (Ausdrucksfeld)	Das Ausdrucks-Listenfeld wird hervorgehoben; es enthält den Standardwert a. Geben Sie `"Test1"` ein und klicken Sie OK an. (Sie müssen die Anführungszeichen verwenden, um den Eintrag als `Text`zeichenfolge zu kennzeichnen. Wenn Sie `Test1` ohne Anführungszeichen eingeben, interpretiert VEE diese Angabe als den Namen eines Anschlusses oder als Name einer globalen Variable.)
WRITE	Verwenden Sie den Standardwert `WRITE`.
TEXT	Verwenden Sie den Standardwert `TEXT`. Die Codierung `TEXT` sendet die Daten als ASCII-Zeichen.
DEFAULT FORMAT	Verwenden Sie `DEFAULT FORMAT`. `DEFAULT FORMAT` verwendet ein entsprechendes VEE-Format wie beispielsweise `STRING`.
EOL ON	Verwenden Sie den Standardwert. Die Standard-`EOL`-Sequenz ist das Escape-Zeichen für eine neue Zeile (`\n`).

3. Klicken Sie `OK` an, um zum Objekt `To File` zurückzukehren. Die Transaktionsleiste sollte jetzt die Anweisung `WRITE TEXT "Test1" EOL` enthalten. Diese Transaktion sendet die Zeichenfolge `Test1` an die angegebene Datei.

Senden eines Zeitstempels an eine Datei

Die Funktion `now()` in der Kategorie `Device ⇒ Function & Object Browser ⇒ Time & Date` liefert die aktuelle Uhrzeit als `Real64 Scalar`. Der Wert der Real-Zahl ist die Anzahl der Sekunden seit 00:00 Uhr am 1. Januar des Jahres 1 n.Chr.

Die Angabe `now()` liefert daher einen Wert von ca. `63G`. VEE liefert dieses Format, weil es mathematisch leicht verarbeitet werden kann und nur wenig Speicherplatz erfordert. Wenn Sie den Zeitstempel in einem leichter lesbaren

Format speichern wollen, verwenden Sie `TIME STAMP FORMAT` im Objekt `To File`. Führen Sie die folgenden Schritte aus, um einen Zeitstempel an eine Datei zu senden.

1. Klicken Sie im gleichen Objekt `To File` den Transaktionsbereich doppelt an, um das Fenster `I/O Transaction` anzuzeigen.

2. Klicken Sie das Eingabefeld "Ausdrucksliste" doppelt an, um das `a` hervorzuheben, und geben Sie `now()` ein. Die Funktion `now()` sendet die aktuelle Uhrzeit aus der Systemuhr im Format `Real`.

3. Ändern Sie das Format `Real` in das `Time Stamp Format`. Klicken Sie den Pfeil neben `DEFAULT FORMAT` an, um das Dropdown-Menü anzuzeigen, und wählen Sie `TIME STAMP FORMAT` aus. Das Dialogfenster `I/O Transaction` zeigt jetzt zusätzliche Einträge an. Legen Sie die Einträge wie folgt fest:

Date & Time	Wählen Sie `Time` im Dropdown-Menü aus.
HH:MM:SS	Klicken Sie die Angabe an, um sie auf `HH:MM` umzuschalten (vom Format Stunde-Minute-Sekunde zum Format Stunde-Minute).
24 HOUR	Klicken Sie die Angabe an, um sie auf `12 HOUR` umzuschalten (vom 24-Stunden-Format auf das Format a.m./p.m.).

Das Dialogfenster "I/O Transaction" sollte jetzt aussehen wie in Abbildung 5-5.

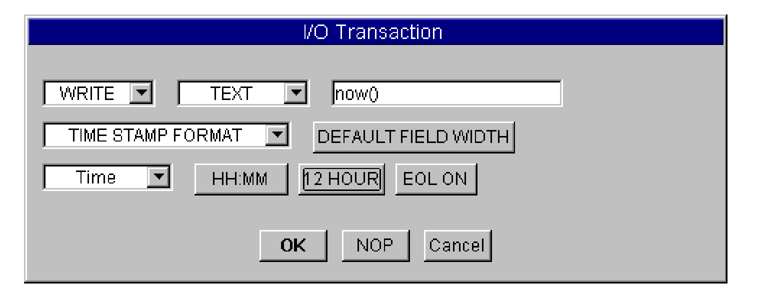


Abbildung 5-5. Das E/A-Transaktionsfeld "TIME STAMP"

4. Klicken Sie OK an, um zum Fenster To File zurückzukehren. Die zweite Transaktionsleiste sollte jetzt die Anweisung WRITE TEXT now() TIME:HM:H12 EOL enthalten.

Senden eines Real Array an eine Datei

Erstellen Sie einen eindimensionalen Array aus vier Elementen mit den Objekten For Count und Collector, und hängen Sie diesen Array an myFile an.

1. Wählen Sie Flow ⇒ Repeat ⇒ For Count aus. Ändern Sie den Standardwert in For Count in den Wert 4.

2. Wählen Sie Data ⇒ Collector aus. Klicken Sie doppelt auf den Collector, um in die offene Ansicht zu wechseln. Verbinden Sie den Datenausgang von For Count mit dem Dateneingang des Collector (der obere Eingangs-Pin). Verbinden Sie den Sequenzausgangs-Pin von For Count mit dem Pin XEQ (dem unteren Eingangs-Pin) am Collector. Stellen Sie anschließend den Collector als Symbol dar.

   Der Collector erstellt jetzt den Array [0, 1, 2, 3], den Sie an die Datendatei senden können.

3. Verwenden Sie dieses Objekt To File , und klicken Sie doppelt im Transaktionsbereich. Öffnen Sie im Dialogfenster I/O Transaction das Menü DEFAULT FORMAT und wählen Sie REAL64 FORMAT aus.

   Das Dialogfenster I/O Transaction zeigt weitere Schaltflächen für die Auswahl REAL64 FORMAT an. Sie können alle standardmäßig ausgewählten Optionen beibehalten; sehen Sie sich jedoch die verfügbaren Optionen zur späteren Verwendung an.

4. Klicken Sie OK an, um das Fenster I/O Transaction zu schließen. Die Transaktionsleiste im Objekt To File sollte jetzt die Anweisung WRITE TEXT a REAL64 STD EOL enthalten. Beachten Sie, dass VEE außerdem automatisch einen Eingangsanschluss a hinzufügt.

5. Verbinden Sie den Ausgang vom Collector mit dem Eingang von a des Objekts To File. Das Programm sollte jetzt aussehen wie in

Abbildung 5-6. (Das konfigurierte Feld I/O Transaction wird ebenfalls angezeigt.)

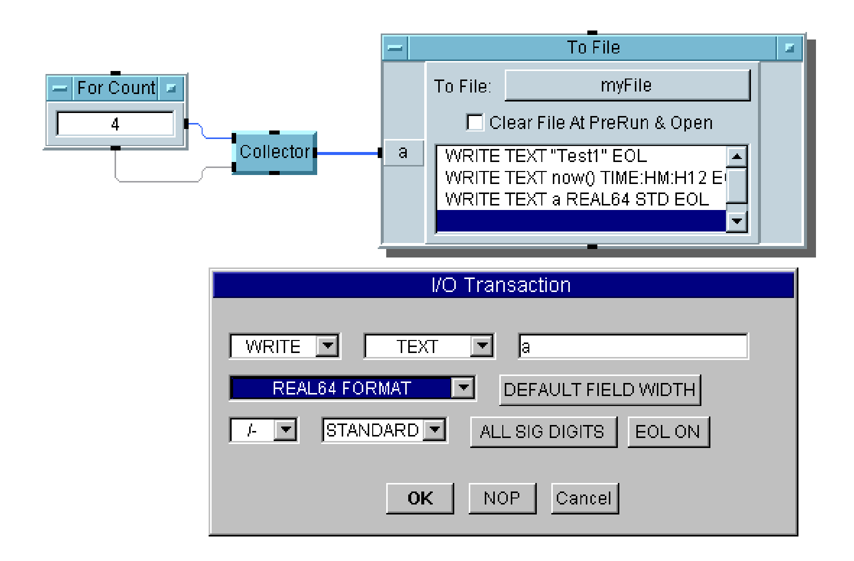


Abbildung 5-6. Speichern von Daten mit dem Objekt "To File"

Abrufen von Daten mit dem Objekt "From File"

Zum Abrufen von Daten mit einem Objekt From File müssen Sie wissen, wie die Daten gespeichert wurden.

Hinweis

Sie können auch Daten mit To DataSet oder From DataSet speichern und abrufen, hierzu brauchen Sie den Typ der Daten in der Datei nicht zu kennen. Eine Beschreibung von Datasets finden Sie im Abschnitt “Verwenden von DataSets zum Speichern und Abrufen von Datensätzen” auf Seite 242

In diesem Beispiel ist der Name eines Tests in einem String Format gespeichert, gefolgt von einem Zeitstempel im Time Stamp Format und einem Array aus Real64-Zahlen. Sie erstellen drei Transaktionen in From File zum Wiedereinlesen der Daten in VEE.

1. Wählen Sie I/O ⇒ From ⇒ File aus und platzieren Sie dieses Objekt unter dem Objekt To File.

2. Verbinden Sie den Sequenzausgangs-Pin des Objekts To File mit dem Sequenzeingangs-Pin des Objekts From File.

   Über diese Sequenzverbindung wird sichergestellt, dass das Objekt To File das Senden von Daten an myFile abgeschlossen hat, bevor From File mit dem Extrahieren von Daten beginnt.

3. Behalten Sie im Objekt From File den Standardwert für die Datendatei myFile bei. Klicken Sie doppelt im Transaktionsbereich, um das Dialogfenster I/O Transaction aufzurufen. Klicken Sie REAL64 FORMAT an und ändern Sie die Angabe in STRING FORMAT, wie in Abbildung 5-7 gezeigt.

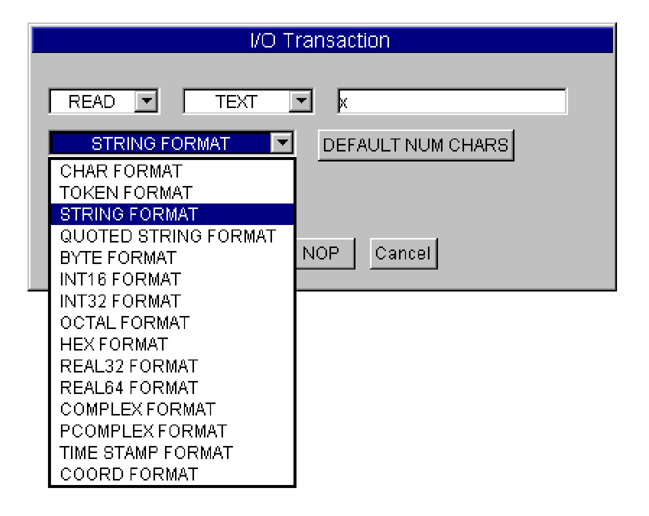


Abbildung 5-7. Auswählen von "String Format"

4. Alle weiteren Standardwerte sind in Ordnung; klicken Sie daher OK an, um das Fenster I/O Transaction zu schließen. Die Transaktionsleiste im Objekt From File sollte jetzt die Anweisung READ TEXT x STR enthalten.

Fügen Sie jetzt zwei weitere Transaktionen hinzu, um den Zeitstempel und den Real-Array wieder einzulesen.

5. Klicken Sie im gleichen Objekt `From File` doppelt unterhalb der ersten Transaktionsleiste. Das Dialogfenster `I/O Transaction` wird angezeigt. Klicken Sie doppelt auf das Eingabefeld für die Ausdrucksliste, um `x` hervorzuheben, und geben Sie `y` ein, damit die zweite Transaktion die Daten wieder an Pin `y` einliest. (Wenn dieser Pin als "`x`" beibehalten wird, überschreibt die zweite Transaktion die Daten, die die erste Transaktion in "`x`" abgelegt hat, statt sie anzuhängen.) Ändern Sie `REAL64 FORMAT` in `STRING FORMAT` und klicken Sie anschließend `OK` an.

Hinweis

Verwenden Sie `STRING FORMAT`, um den Zeitstempel als Textzeichenfolge wieder einzulesen. `TIME STAMP FORMAT` wandelt die Zeitstempeldaten um in eine `Real`-Zahl.

6. Klicken Sie im gleichen Objekt `From File` doppelt unterhalb der zweiten Transaktionsleiste, um das Dialogfenster `I/O Transaction` anzuzeigen. Legen Sie die Einträge wie folgt fest:

(Ausdrucksfeld)	Ändern Sie `x` in `z`, sodass der `Real`-Array am Ausgangsanschluss `Z` wieder eingelesen wird.
SCALAR	Ändern Sie `SCALAR` in `ARRAY 1D`.
SIZE:	Das Fenster `I/O Transaction` fügt jetzt eine Schaltfläche `SIZE` hinzu. In diesem Fall hat der Array vier Elemente. Ersetzen Sie `10` durch `4` und klicken Sie `OK` an.

Hinweis

Wenn Sie die Größe des Array nicht kennen, können Sie `SIZE` in `TO END` umschalten. Dadurch werden die Daten bis zum Ende der Datei gelesen, ohne dass VEE ihre genaue Größe kennt. Sie können diese Funktion beispielsweise verwenden, um den gesamten Inhalt einer Datei als Zeichenfolge-Array einzulesen, um den Inhalt zu überprüfen.

Die Transaktionsleiste im Objekt `From File` sollte jetzt die Anweisungen `READ TEXT y STR` und `READ TEXT z REAL64 ARRAY:4` enthalten. Beachten Sie, dass VEE automatisch die Datenausgangsanschlüsse für x, y und z hinzufügt. Sie können auch im Objektmenü über ⇒ `Add Terminal` und `Delete Terminal` oder mit den Direktaufrufen **Strg-A** und **Strg-D** Ein- und Ausgangsanschlüsse hinzufügen.

7. Wählen Sie `Display` ⇒ `AlphaNumeric` aus und klonen Sie dieses Objekt zwei Mal, um drei Anzeigen zu erhalten. Verbinden Sie die Objekte `AlphaNumeric` mit den drei Datenausgangs-Pins an `From File`. Vergrößern Sie die Array-Anzeige durch Anklicken und Ziehen des Objekts an einer beliebigen Ecke.

   Tip: Sie können auch die Größe der `AlphaNumeric`-Anzeigen ändern, indem Sie bei der ersten Auswahl der Objekte im Menü die Objektumrisse anklicken und ziehen.

8. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 5-8.

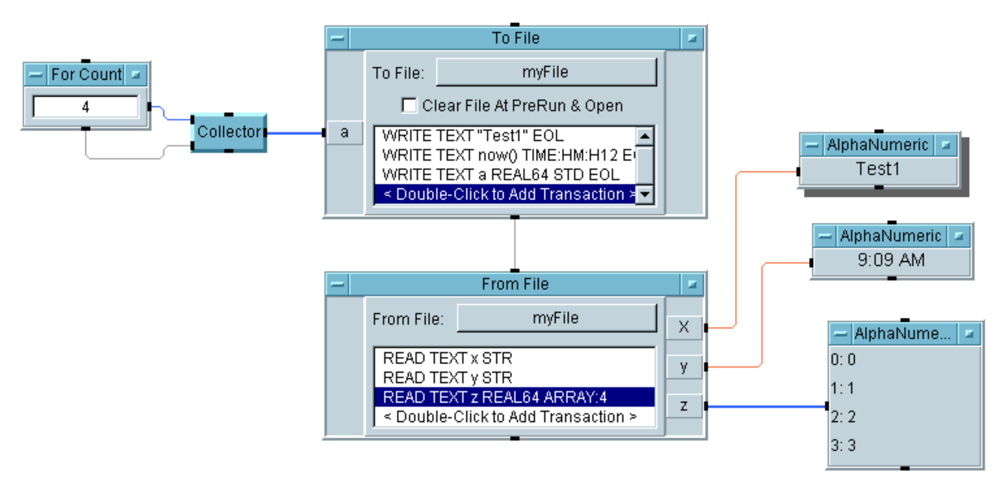


Abbildung 5-8. Abrufen von Daten mit dem Objekt "From File"

Beachten Sie, dass das erste Alphanumeric-Objekt den Titel anzeigt, das zweite den Zeitpunkt des Tests und das dritte die Zahlen in dem Array.

Verwenden von Datensätzen zum Speichern gemischter Datentypen

Der Datentyp "Record" (Datensatz) kann verschiedene Datentypen in einem einzigen Datencontainer speichern. Der Datensatz kann jeden beliebigen VEE-Datentyp enthalten. Die Daten können die Form eines Skalars oder eines Array haben. Sie können einen Testnamen, einen Zeitstempel und einen Real-Array in einer einzigen Datenstruktur speichern.

Die einzelnen Elemente in einem Datensatz sind als Felder gespeichert und werden über eine Punktnotation aufgerufen. `Rec.Name` greift beispielsweise auf das Feld `Name` in einem Datensatz mit dem Namen `Rec` zu. In einem Array von Records kennzeichnet `Rec[2].Name` das Feld `Name` im dritten Datensatz des Array. Die Indexierung aller Arrays beginnt bei Null.

Die Strukturierung der Testdaten mit dem Datentyp "Record" hat verschiedene Vorteile:

- Sie können logische Gruppierungen gemischter Datentypen in einem einzigen Container erstellen; die Entwicklung und Pflege des Programms wird dadurch vereinfacht. Sie können beispielsweise die folgenden Felder für einen Datensatz zum Speichern von Testdaten verwenden: test name, value returned, pass or fail indicator, time stamp, nominal value expected, upper pass limit, lower pass limit sowie eine Beschreibung des Tests.

- Sie brauchen auf diese Weise nur einen einzigen Datencontainer zu bearbeiten statt acht separater Container. Das Programm wird dadurch einfacher und übersichtlicher.

- Sie können Datensätze in DataSets in VEE speichern und daraus abrufen. Ein **DataSet** ist eine spezielle Datei, die zum Speichern von Datensätzen erstellt wurde. Wenn Sie Records aus einem DataSet abrufen, brauchen Sie die Datentypen nicht zu kennen. VEE bietet Objekte zum Abrufen, Sortieren und Suchen der in DataSets gespeicherten Informationen.

Übung 5-4: Verwenden von Datensätzen

In dieser Übung wird die Verwendung des Datentyps "Record" (Datensatz) beschrieben. Sie lernen, wie Sie einen Datensatz erstellen, wie Sie ein bestimmtes Feld in diesem Datensatz abrufen, wie Sie ein ausgewähltes Feld einstellen und wie Sie den gesamten Datensatz in einem einzigen Arbeitsschritt auflösen. Diese Übung verwendet auch die Zeitstempelfunktion `now()` in einer anderen Weise.

Erstellen eines Datensatzes

Erstellen Sie einen Datensatz mit drei Feldern: dem als `String` gespeicherten Namen eines Tests, einem als `Real Scalar` gespeicherten Zeitstempel und einem simulierten Testergebnis, das als vier Elemente in einem `Array of Reals` gespeichert ist. Wenn Sie diese Felder in der nächsten Übung abrufen, werden Sie sehen, dass Sie den Zeitstempel in eine Reihe verschiedener Anzeigeformate umwandeln können.

1. Erstellen Sie den Testnamen durch Auswählen von `Data` ⇒ `Constant` ⇒ `Text` und Eingeben von `Test1` im Eingabefeld. Benennen Sie das Objekt um in `Text Constant`. Stellen Sie `Text Constant` als Symbol dar.

2. Wählen Sie `Device` ⇒ `Function & Object Browser` aus. Klicken Sie `Built-in Functions` unter `Type` und `Time & Date` unter `Category` an, wählen Sie `now` unter `Functions` aus und klicken Sie `Create Formula` an. Platzieren Sie das Objekt unter `Text Constant`.

3. Wählen Sie `Data` ⇒ `Constant` ⇒ `Real64` aus und platzieren Sie das Objekt unter `now()`.

   Sie können dieses `Scalar Real64`-Objekt in einen `Array 1D` umwandeln, indem Sie `Properties...` im `Real64`-Objektmenü anklicken und `1D Array` auswählen.

4. Öffnen Sie das Fenster `Constant Properties`, indem Sie doppelt auf die `Real64`-Titelleiste klicken. Wählen Sie `1D Array` unter `Configuration` aus, ändern Sie `Size` in `4` und klicken Sie `OK` an.

Geben Sie vier Werte in diesen Array ein, indem Sie neben dem Element `0000` doppelt klicken, um den ersten Eintrag hervorzuheben und anschließend die Werte `2.2`, `3.3`, `4.4`, `5.5` eingeben. Wechseln Sie mit der **Tabulatortaste** zu den einzelnen Einträgen. Zeigen Sie `Real64` als Symbol an.

5. Wählen Sie `Data` ⇒ `Build Data` ⇒ `Record` aus und platzieren Sie das Objekt rechts von den drei anderen Objekten. Fügen Sie einen dritten Dateneingangsanschluss hinzu, sodass Sie Daten über drei Felder eingeben können. Öffnen Sie die einzelnen Anschlüsse, indem Sie doppelt auf dem Anschluss klicken, und benennen Sie die drei Eingangsanschlüsse um in `testname`, `time` und `data`.

   Das Feld `Output Shape` (Ausgabeform) des Objekts `Build Record` kann umgeschaltet werden zwischen `Scalar` und `Array`. Der Standardwert `Scalar` ist für die meisten Situationen die richtige Wahl. (Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.)

6. Verbinden Sie das Objekt `Text Constant` mit dem Anschluss `testname`, das Objekt `now()` mit dem Anschluss `time` und das Objekt `Real64` mit dem Datenanschluss am Objekt `Build Record`.

7. Führen Sie das Programm aus. Klicken Sie doppelt auf den Datenausgangsanschluss von `Record`, um den Datensatz zu prüfen. Er sollte aussehen wie in Abbildung 5-9.

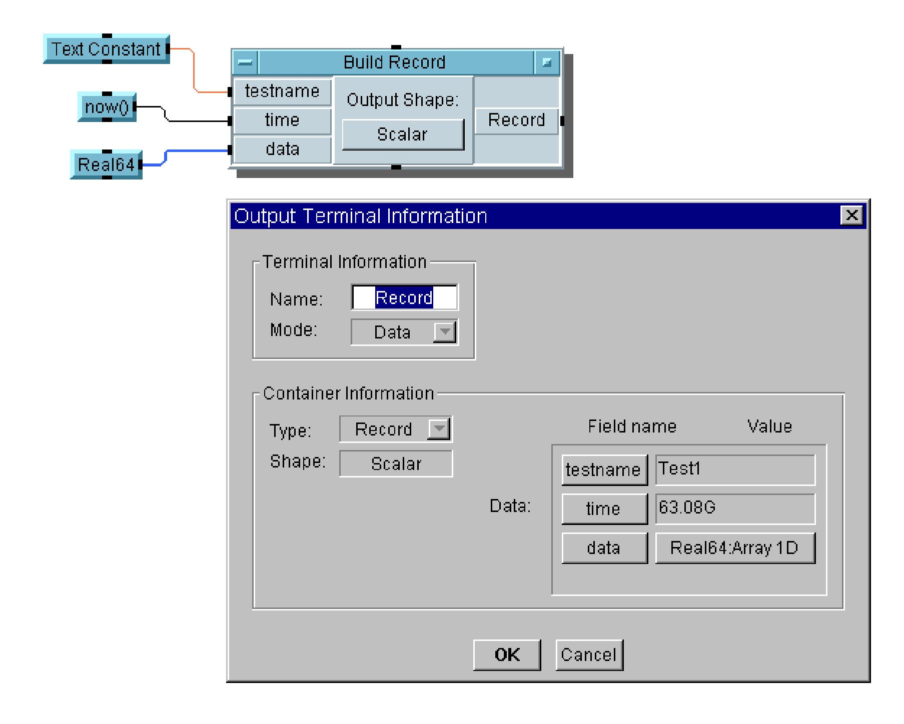


Abbildung 5-9. Informationen zum Ausgangsanschluss an einem Datensatz

Sie sehen die drei Felder und ihre Werte. Wenn Sie die Schaltfläche `Real64: Array 1D` anklicken, zeigt ein Listenfeld die tatsächlichen Werte an. Beachten Sie, dass der Zeitstempel als `Real64 Scalar` gespeichert wurde. In der nächsten Übung wandeln Sie den Zeitstempel in eine leichter lesbare Form um. Klicken Sie `OK` an, um das Dialogfenster `Output Terminal Information` zu schließen. Speichern Sie das Programm als `records.vee`.

Abrufen eines Felds aus einem Datensatz

Verwenden Sie das Objekt `Get Field` zum Extrahieren der drei Felder aus dem Datensatz, und zeigen Sie anschließend die Werte für die Felder an.

1. Öffnen Sie das Programm `records.vee`.

2. Wählen Sie `Data ⇒ Access Record ⇒ Get Field` aus. Das Objekt erscheint mit dem Titel `rec.field`.

   Der Dateneingang mit der Beschriftung `rec` verwendet einen Datensatz unabhängig von der Anzahl und dem Typ der Felder. `Rec.field` ist der Standardwert im Eingangsfeld; Sie können diese Angabe jedoch ändern, um ein beliebiges Feld abzurufen. `Rec` bezieht sich auf den Datensatz am Dateneingangsanschluss dieses Namens. (Denken Sie daran, dass VEE *nicht* zwischen Groß- und Kleinbuchstaben unterscheidet.)

Hinweis Das Objekt `Get Field` ist eine mit Eingängen und einem Ausdruck konfigurierte `Formula` wie die Formeln im `Function & Object Browser`.

3. Klonen Sie `rec.field` zwei Mal und platzieren Sie die Objekte rechts von `Build Record`.

4. Verbinden Sie den Datenausgang von `Build Record` mit allen drei Objekten `rec.field`.

   Da die drei Felder als `testname`, `time` und `data` gespeichert wurden, müssen Sie die Objekte `rec.field` ändern, um das entsprechende Feld zu erhalten.

5. Ändern Sie die drei Objektausdrucksfelder `rec.field` in `rec.testname`, `rec.time` und `rec.data`.

6. Wählen Sie `Display ⇒ AlphaNumeric` aus und klonen Sie dieses Objekt zwei Mal. Verbinden Sie die drei Anzeigen mit den drei Objekten `rec.field`. Ändern Sie die Größe der dritten Anzeige, sodass sie genügend Platz für den Real-Array bietet (ungefähr dreimal so lang wie die anderen Objekte).

7. Öffnen Sie das Objektmenü der zweiten `AlphaNumeric`-Anzeige und wählen Sie `Properties` und anschließend den Ordner `Number` aus. Klicken Sie links von `Global Format`, um das Häkchen zu entfernen.

   Legen Sie das Anzeigeformat fst. Öffnen Sie das Menü `Standard` im Abschnitt `Real` . Wählen Sie `Time Stamp` aus und klicken Sie `OK` an.

8. Klicken Sie `HH:MM:SS` an, um auf `HH:MM` umzuschalten. Klicken Sie `24 HOUR` an, um auf `12 HOUR` umzuschalten. Siehe Abbildung 5-10.

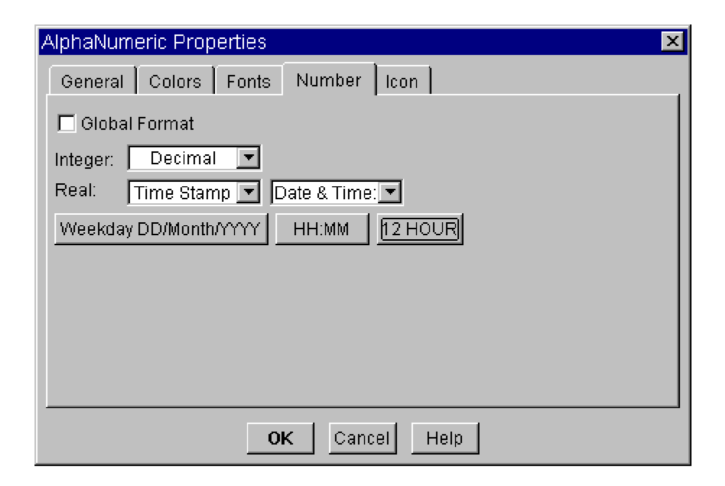


Abbildung 5-10. Feld "AlphaNumeric Properties"

9. Führen Sie das Programm aus und speichern Sie es als `getfield.vee`. Das Programm sollte aussehen wie in Abbildung 5-11.

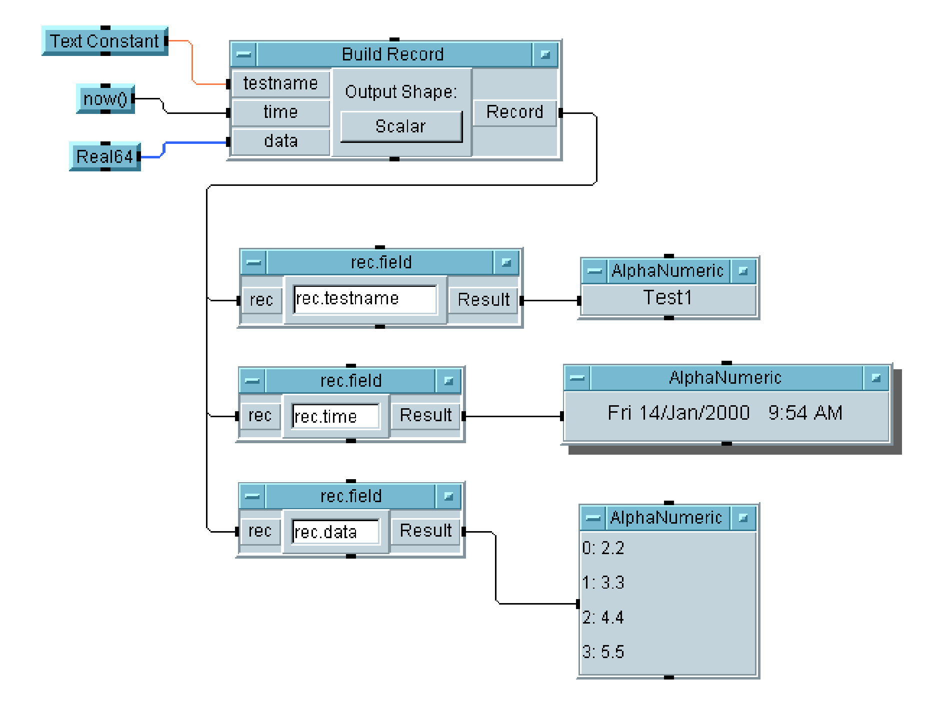


Abbildung 5-11. Verwenden des Objekts "Get Field"

Beachten Sie, dass die zweite Anzeige den Wochentag, das Datum und die Uhrzeit in Stunden, Minuten und der Angabe "a.m./p.m." (Vor- oder Nachmittag) auflistet.

Einstellen eines Felds in einem Datensatz

Diese Übung zeigt, wie Daten in spezifischen Feldern eines Datensatzes geändert werden.

Hinweis

Sie können diesen `Record` mit verschiedenen Tests wieder verwenden.

1. Öffnen Sie das Programm `getfield.vee`.

2. Löschen Sie alle Objekte nach `Build Record`, indem Sie die Objekte auswählen und **Strg-X** drücken.

1. Wählen Sie `Data ⇒ Access Record ⇒ Set Field` aus und platzieren Sie das Objekt rechts von `Build Record`. Verbinden Sie den Ausgang von `Build Record` mit dem Eingang `rec` des Objekts `Set Field`. Der Titel lautet `rec.field = b`.

   `Set Field` funktioniert durch Zuordnen des Ausdrucks auf der rechten Seite des Zuordnungssymbols (`=`) zur linken Seite. Das angegebene Feld von `rec` wird daher so geändert, dass es Werte von der rechten Seite enthält. Der Rest des Datensatzes bleibt unverändert. Sie verbinden den ankommenden Datensatz mit `rec` und den ankommenden neuen Wert mit `b`. Der geänderte Datensatz wird an dem Datenausgangsanschluss `rec` angelegt.

Hinweis

Das Objekt `Set Field` ist eine mit Eingängen und einem Ausdruck konfigurierte `Formula` wie die Formeln um `Function & Object Browser`.

2. Ändern Sie den Ausdruck in `rec.data[*]=b`, um den Wert des Vier-Element-Arrays im Datenfeld zu ändern. (Sie müssen die Array-Notation `[*]` verwenden, da Sie den gesamten Array im Feld dieses Datensatzes ändern.) Die neuen Werte für den Array werden am Eingangsanschluss `b` angelegt.

3. Wählen Sie `Data ⇒ Constant ⇒ Real64` aus und platzieren Sie dieses Objekt unter dem Objekt `Build Record`. Öffnen Sie das Objektmenü und wählen Sie `Properties` aus. Wählen Sie `1D Array` unter `Configuration` aus, ändern Sie `Size` in `4` und klicken Sie `OK` an.

   Wenn die neuen Werte für das Datensatzfeld in einem Array enthalten sind, muss dieser die gleiche Größe wie der aktuelle Array haben.

   Geben Sie die Werte `1`, `2`, `3`, `4` in `Real64` ein, indem Sie den ersten Eintrag hervorheben und mit der **Tabulatortaste** zu den weiteren Einträgen springen. (Drücken Sie nach dem letzten Eintrag nicht die **Tabulatortaste**.) Verbinden Sie Real64 mit dem Eingang `b` von `Set Field` (mit dem Titel `rec.field=b`).

Verwenden Sie jetzt das Objekt `Get Field` zum Extrahieren des Felds `rec.data` aus dem Datensatz und zeigen Sie die Ergebnisse an.

4. Wählen Sie `Data ⇒ Access Record ⇒ Get Field` aus und platzieren Sie das Objekt unter dem Objekt `Set Field` (`rec.field=b`). Ändern Sie den `Get Field`-Objektausdruck von `rec.field` in `rec.data`. Verbinden Sie den Datenausgang von `rec.field = b` mit dem Dateneingang von `rec.field`.

Hinweis Sie können auch ein `Formula`-Objekt mit `A.data` im Ausdrucksfeld verwenden.

5. Wählen Sie eine `AlphaNumeric`-Anzeige aus, ändern Sie ihre Größe entsprechend dem Array und verbinden Sie sie mit dem Ausgangs-Pin von `rec.field`.

6. Führen Sie das Programm aus und speichern Sie es als `setfield.vee`. Das Programm sollte aussehen wie in Abbildung 5-12.

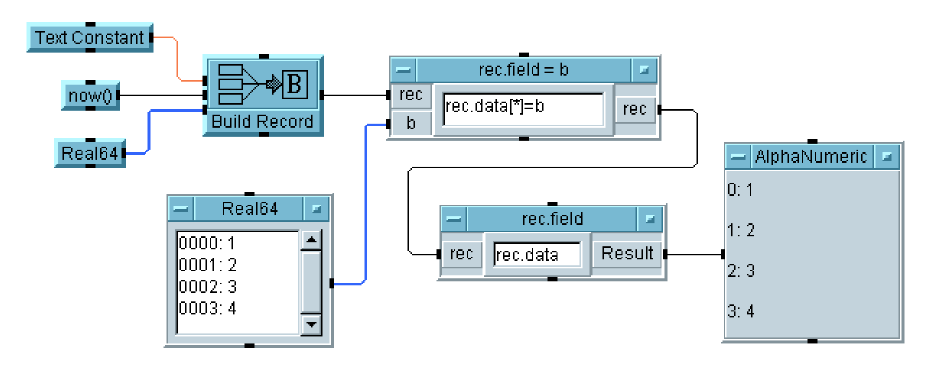


Abbildung 5-12. Verwenden des Objekts "Set Field"

Sie können jedes beliebige `Record`-Feld wie in diesem Beispiel gezeigt ändern. Außerdem können Sie Felder auch teilweise ändern. Versuchen Sie beispielsweise, den Ausdruck in `rec.field = b` in den Ausdruck `rec.data[1]=20` zu ändern. Löschen Sie anschließend den Eingang `b` von `rec.field = b`. Führen Sie das Programm erneut aus. Der Array sollte wie folgt angezeigt werden: `2.2`, `20`, `4.4`, `5.5`.

Datensatz in einem einzigen Arbeitsschritt auflösen

Zum Extrahieren aller Datensatzfelder und zum Abrufen einer Liste von Feldnamen und ihrer Typen verwenden Sie das Objekt `UnBuild Record`.

1. Öffnen Sie das Programm `setfield.vee`. Löschen Sie alle Objekte nach `Build Record`.

2. Wählen Sie `Data` ⇒ `UnBuild Data` ⇒ `Record` aus, platzieren Sie dieses Objekt unter `Build Record`, wechseln Sie zur offenen Ansicht und verbinden Sie den Ausgang von `Build Record` mit dem Eingang von `UnBuild Record`. Fügen Sie `UnBuild Record` einen weiteren Datenausgangs-Pin hinzu und benennen Sie seine Ausgänge `A`, `B` und `C` in die Feldnamen um: `testname`, `time` und `data`.

3. Wählen Sie eine `AlphaNumeric`-Anzeige aus und klonen Sie sie vier mal. Verbinden Sie die fünf Anzeigen mit den fünf Ausgangsanschlüssen an `UnBuild Record`. Sie müssen die Anzeigen für `Name List`, `Type List` und `Data` vergrößern, damit genügend Platz für die Arrays zur Verfügung steht. Konfigurieren Sie außerdem die Anzeige von time auf die aktuelle Zeit im Format Tag/Monat/Jahr und Stunde, Minute im 12-Stunden-Format um.

4. Führen Sie das Programm aus und speichern Sie es als `unbuild.vee`. Es sollte aussehen wie in Abbildung 5-13.

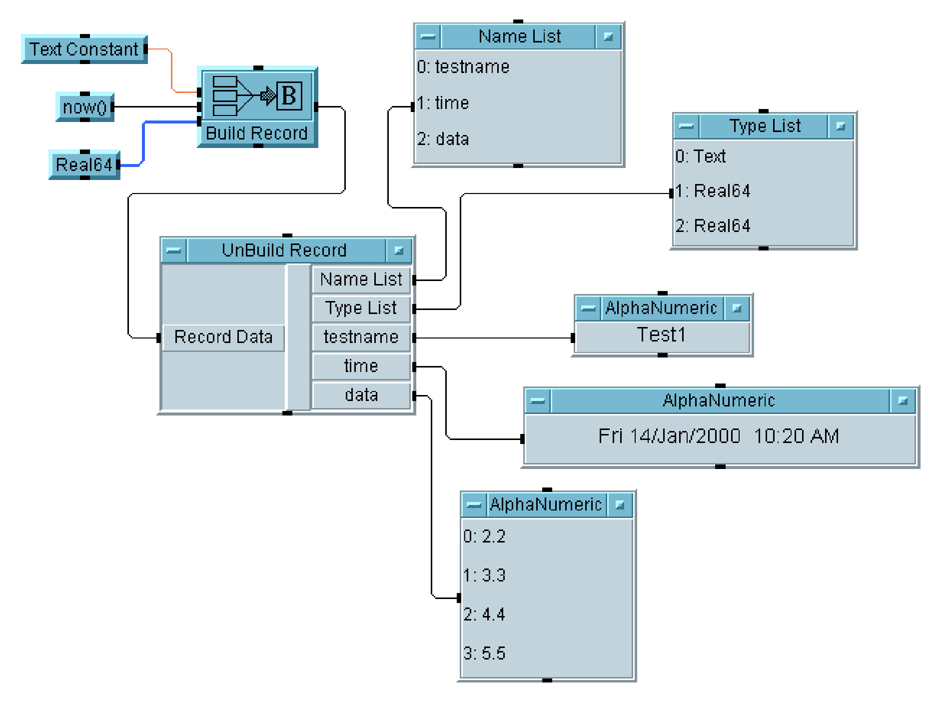


Abbildung 5-13. Verwenden des Objekts "UnBuild Record"

Beachten Sie, dass der Pin `Name List` die Namen der drei Felder `testname`, `time` und `data` des Datensatzes liefert; `Type List` kennzeichnet `testname` als Typ `Text` sowie `time` und `data` als Typ `Real64`.

Verwenden von DataSets zum Speichern und Abrufen von Datensätzen

`DataSets` können ein oder mehrere Datensätze abrufen. VEE-Objekte entpacken die Datensätze. Durch das Speichern von Datensätzen in DataSets statt in Dateien brauchen Sie sich die Datentypen daher nicht zu merken. Sie können auch Sortier- und Suchoperationen mit den Daten ausführen und auf diese Weise Ihre eigene angepasste Testdatenbank erstellen.

Übung 5-5: Verwenden von DataSets

Ein `DataSet` ist einfach ein Array von `Records`, die in einer Datei gespeichert wurden. Diese Übung zeigt, wie die Daten in einem `DataSet` gespeichert und von dort wieder abgerufen werden.

Speichern und Abrufen eines Datensatzes aus einem DataSet

In dieser Übung wird ein Array aus zehn Datensätzen erstellt; jeder dieser Datensätze enthält drei Felder mit einem Testnamen, einem `Real64 Scalar` und einem Array von `Real`-Zahlen. Der Array von `Records` wird in einem `DataSet` gespeichert; anschließend werden die Datensätze abgerufen und angezeigt.

1. Wählen Sie `Flow ⇒ Start` aus. Wählen Sie `Flow ⇒ Repeat ⇒ For Count` aus und platzieren Sie das Objekt unter `Start`. Wählen Sie `Device ⇒ Formula` aus und platzieren Sie das Objekt rechts von `For Count`. Verbinden Sie `Start` mit dem Sequenzeingangs-Pin an `For Count`; verbinden Sie den Datenausgangs-Pin von `For Count` mit dem Dateneingangs-Pin von `Formula`.

2. Klicken Sie doppelt auf das Ausdrucksfeld `Formula`, um den Standardausdruck hervorzuheben, und geben Sie anschließend `"test" + a` ein.

   Wenn Sie `Start` anklicken, gibt das Objekt `For Count` sequenziell ganze Zahlen von Null bis Neun an den Pin `A` von `Formula` aus. Im Objekt `Formula` werden die ganzen Zahlen dem Wort `"test"` hinzu-

gefügt und als `Text Scalars` ausgegeben: `test0`, `test1`, `test2`,...,`test9`. Diese Werte bilden die ersten Felder in den zehn Datensätzen.

3. Wählen Sie `Data` ⇒ `Build Data` ⇒ `Record` aus und platzieren Sie das Objekt rechts von `Formula`. Fügen Sie einen Dateneingangs-Pin hinzu. Verbinden Sie den Datenausgang von `Formula` mit dem Eingang `A` von `Build Record`.

4. Wählen Sie das Symbol **Function & Object Browser** in der Symbolleiste aus.

    a. Wählen Sie `Built-in Functions`, `Probability & Statistics` und `random` aus, um das Objekt `random (low, high)` zu erstellen. Platzieren Sie das Objekt unter dem Objekt `Formula`.

    b. Löschen Sie die Eingangsanschlüsse und ändern Sie die Eingangsparameter von `low` in `0` sowie von `high` in `1`.

    c. Benennen Sie das Objekt `Random Number` um und verbinden Sie seinen Datenausgang mit dem Anschluss `B` von `Build Record`.

5. Verbinden Sie den Sequenzausgangs-Pin des Objekts `Formula` mit dem Sequenzeingangs-Pin von `Random Number`. Durch das Verbinden der Sequenz-Pins wird sichergestellt, dass jede Iteration des Programms eine neue Zufallszahl in das Feld `B` des jeweiligen Datensatzes stellt.

6. Wählen Sie `Data` ⇒ `Constant` ⇒ `Real64` aus. Platzieren Sie das Objekt `Real64` unter dem Objekt `Formula`.

    a. Öffnen Sie das Objektmenü und klicken Sie `Properties` an. Geben Sie `Real Array` als Titel ein, klicken Sie unter `Configuration` die Option `1D Array` an und ändern Sie `Size` in `3`. Klicken Sie `OK` an.

    b. Heben Sie alle Einträge in dem Array hervor, indem Sie doppelt darauf klicken, und geben Sie die Zahlen `1`, `2` und `3` ein.

    c. Verbinden Sie den Datenausgang des `Real Array` mit dem Anschluss `C` des Objekts `Build Record`.

7. Wählen Sie `I/O ⇒ To ⇒ DataSet` und platzieren Sie dieses Objekt unter `Build Record`. Verbinden Sie den Datenausgang von `Build Record` mit seinem Dateneingang. Behalten Sie den Standarddateinamen `myfile` bei und markieren Sie `Clear File At PreRun`.

8. Führen Sie das Programm aus. Das Programm sollte einen Array aus zehn Datensätzen in den Dataset `myFile` stellen (siehe Abbildung 5-14).

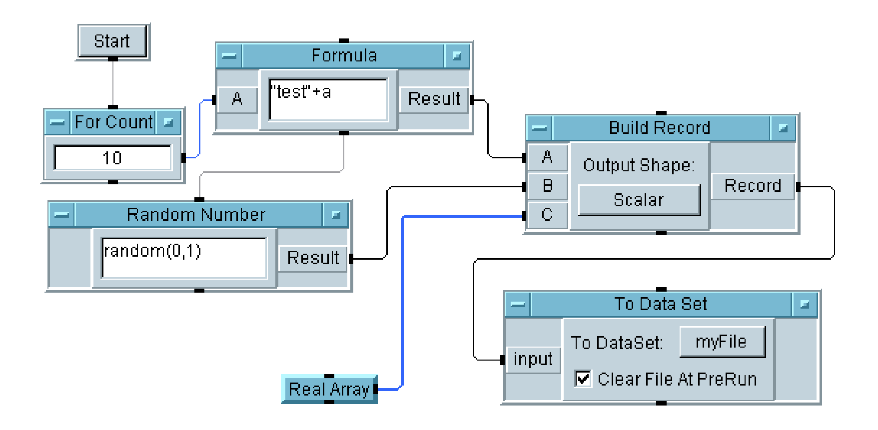


Abbildung 5-14. Speichern eines Array von Datensätzen in einem DataSet

Rufen Sie jetzt den Array von Datensätzen ab und zeigen Sie ihn mit den Objekten `From DataSet` und `Record Constant` an.

9. Wählen Sie `I/O ⇒ From ⇒ DataSet` aus und platzieren Sie dieses Objekt unter `For Count`. Behalten Sie den Standarddateinamen `myFile` bei. Klicken Sie das Feld `Get Records` an, um es von `One` auf `All` umzuschalten. Behalten Sie schließlich den Standardwert `1` im Ausdrucksfeld am unteren Rand bei.

   Mit diesen Einstellungen ruft VEE den DataSet in `myFile` auf und findet alle Datensätze, die dem Kriterium in dem Ausdrucksfeld entsprechen. Wenn Sie `Get Records` auf `One` setzen, gibt VEE zunächst den ersten Datensatz aus, der dem Kriterium in dem Ausdrucksfeld entspricht. Die `1` kennzeichnet eine `TRUE`-Bedingung; dies bedeutet, dass alle Datensätze dem Kriterium entsprechen, sodass der gesamte Array von Daten-

sätzen in der Datei an den Ausgangs-Pin `Rec` angelegt wird. Weitere Anwendungsmöglichkeiten des Ausdrucksfelds werden in anderen Übungen beschrieben. Schlagen Sie weitere Informationen unter `Help` im Objektmenü nach.

Verbinden Sie den Sequenzausgangs-Pin von `For Count` mit dem Sequenzeingang am Objekt `From Data Set`. Auf diese Weise wird sichergestellt, dass der Teil des Programms, der Daten an `myFile` sendet, ausgeführt wird, bevor die Daten aus der Datei gelesen werden. Sie können `Show Data Flow` einschalten, um die Reihenfolge der Ereignisse anzuzeigen.

10. Wählen Sie `Data` ⇒ `Constant` ⇒ `Record` aus und platzieren Sie dieses Objekt unter `To Data Set`. Öffnen Sie das Objektmenü und wählen Sie `Add Terminal` ⇒ `Control Input` aus. Klicken Sie `Default Value` in dem angezeigten Listenfeld an und klicken Sie `OK` an. Vergrößern Sie das Objekt `Record`, sodass Sie bei der Ausführung des Programms die Ergebnisse sehen können.

    Der empfangene Datensatz wird zum Standardwert. In diesem Fall empfängt `Record` einen Array von Datensätzen von dem Objekt `From Data Set` und formatiert sich selbst, um diesen Array von Datensätzen anzuzeigen.

11. Verbinden Sie den Ausgangs-Pin `Rec` von `From Data Set` mit dem Pin `Default Value` an `Record`. Wenn Sie diesen Anschluss anzeigen wollen, öffnen Sie das Objektmenü und wählen Sie `Properties`, dann `Show Terminals` und schließlich `OK` aus. Es erscheint eine gepunktete Linie zwischen `From Data Set` und `Record`.

Hinweis Eine gepunktete Linie zwischen zwei Objekten kennzeichnet eine Steuerlinie.

12. Führen Sie das Programm aus und speichern Sie es als `dataset1.vee`. Das Programm sollte aussehen wie in Abbildung 5-15.

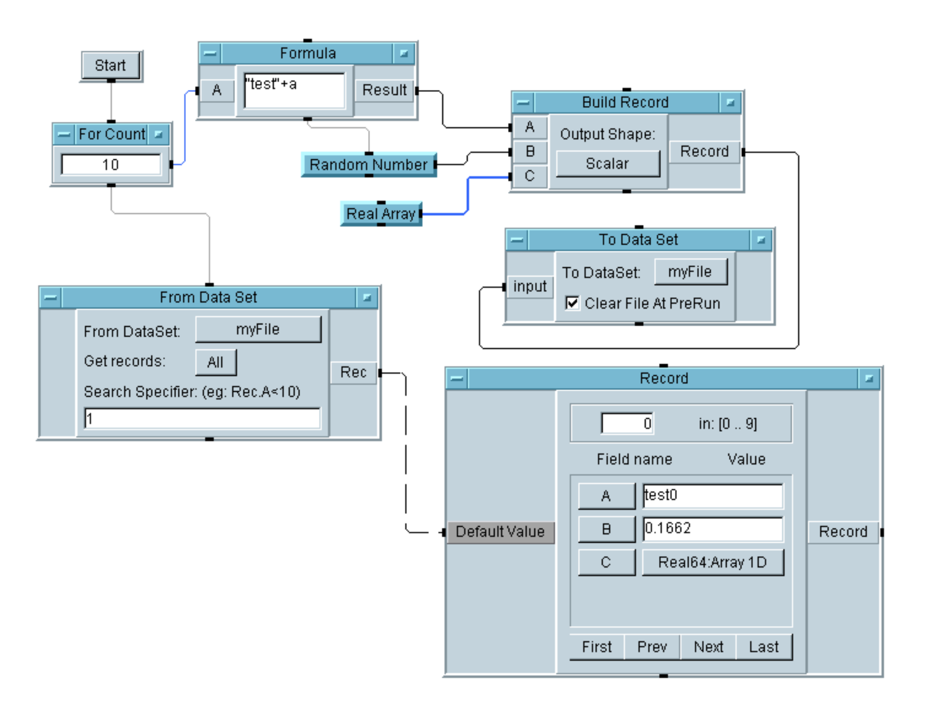


Abbildung 5-15. Speichern und Abrufen von Daten mit DataSets

Hinweis

Ein Objekt `From Data Set` muss mindestens einen Datensatz enthalten, der dem Kriterium entspricht; ansonsten gibt VEE eine Fehlermeldung aus. Um einen Fehler zu vermeiden, fügen Sie dem Objekt einen Ausgangs-Pin `EOF` (end-of-file) hinzu; dieser Pin wird aktiviert, wenn kein Datensatz das Kriterium erfüllt. Sie können dann dem Programm weitere Aktionen hinzufügen für den Fall, dass `EOF` auslöst.

Anpassen einer einfachen Testdatenbank

Sie können in einem DataSet Informationen suchen und sortieren, z. B. Testnamen, Zeitstempel, Testparameter, Testwerte, Kennzeichen für eine erfolgreiche oder fehlgeschlagene Ausführung sowie Testbeschreibungen. DataSet-Datensätze können somit als Testdatenbank verwendet werden. Zum Suchen von Informationen können Sie das Objekt From Data Set wie folgt verwenden:

- Das Ausdrucksfeld im Objekt From Data Set wird für Suchoperationen verwendet.
- Die Funktion sort() kann zum Sortieren von Datensätzen nach einem angegebenen Feld verwendet werden.

Übung 5-6: Verwenden von Such- und Sortieroperationen mit DataSets

In dieser Übung lernen Sie, wie Sie einen DataSet nach Informationen durchsuchen, eine Bedienerschnittstelle für die Suchoperation erstellen und eine Sortieroperation programmieren.

Durchführen einer Suchoperation mit DataSets

1. Öffnen Sie das Programm dataset1.vee.
2. Klicken Sie doppelt auf das Ausdrucksfeld am unteren Rand des Objekts From Data Set, um den aktuellen Ausdruck 1 hervorzuheben. Geben Sie Rec.B>=0.5 ein. Das Objekt gibt jetzt alle Datensätze aus, wobei Feld B (die Zufallszahl in unserem Code) größer oder gleich 0.5 ist.
3. Fügen Sie einen EOF-Pin hinzu, der auslöst, wenn keine Datensätze dem Kriterium im Ausdrucksfeld entsprechen. Positionieren Sie den Cursor auf den Datenausgangsbereich des Objekts From Data Set und drücken Sie **Strg-A**. Ein EOF-Ausgangs-Pin wird dem Objekt From Data Set hinzugefügt (siehe Abbildung 5-16).

Hinweis

Zum Hinzufügen eines `EOF`-Pin könnten Sie auch das Objektmenü öffnen und `Add Terminal` ⇒ `Data Output...` anklicken.

4. Führen Sie das Programm aus und speichern Sie es als `dataset2.vee`.

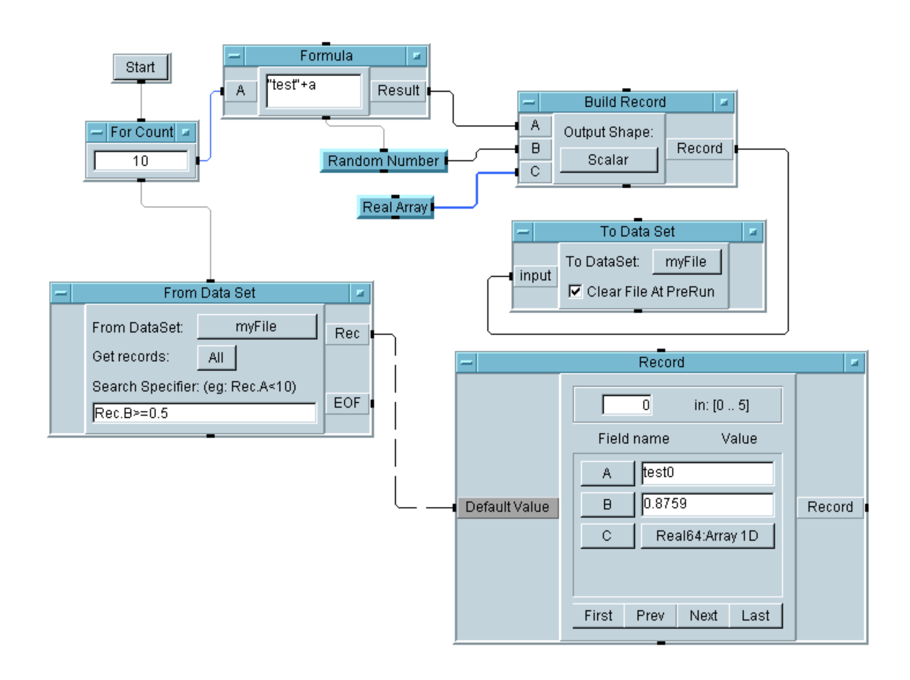


Abbildung 5-16. Eine Suchoperation mit DataSets

Erstellen einer Bedienerschnittstelle für eine Suchoperation

In dieser Übung wird ein Menü für einen Bediener hinzugefügt, um Daten aus der Datenbank der Testergebnisse zu extrahieren. Die Bedienerschnittstelle ist gesichert, um versehentliche Änderungen an dem Programm zu verhindern.

Die Spezifikation des Programms lautet wie folgt:

- Bereitstellen eines Testmenüs, das es dem Bediener ermöglicht, einen bestimmten Test aus `test0` bis `test9` auszuwählen, aus dem er alle entsprechenden Testdaten abrufen kann.
- Anzeigen der angegebenen Testergebnisse mit den beschrifteten Feldern und Werten. Der Bediener soll mit der Anzeige interagieren können, um ausführlichere Informationen abzurufen.
- Einbeziehen klarer Bedienungshinweise.

Führen Sie zum Erstellen des Programms die folgenden Schritte durch:

1. Öffnen Sie das Programm `dataset2.vee`.

Fügen Sie einen Steuereingang hinzu, der die programmgesteuerte Eingabe des Ausdrucks im Objekt `From Data Set` ermöglicht.

2. Öffnen Sie das `From Data Set` -Objektmenü und wählen Sie `Add Terminal...` ⇒ `Control Input...` aus. Wählen Sie `Formula` in dem angezeigten Menü aus. Ein `Formula`-Eingangsanschluss wird angezeigt. Klicken Sie das Feld `Get records` an, um von `All` auf `One` umzuschalten, damit ein Testdatensatz nach dem anderen aufgerufen wird.

   Der Bediener soll einen bestimmten Testnamen auswählen können. Die Testnamen befinden sich im Feld `A` aller Datensätze. Fügen Sie den folgenden Ausdruck hinzu:

   `Rec.A==`<*Testname in Anführungszeichen*>

   `Rec.A` gibt den Datensatz aus, in dem Feld `A` dem vom Bediener ausgewählten Testnamen entspricht. Wenn der Bediener beispielsweise `test6` auswählt, muss der Ausdruck lauten: `Rec.A=="test6"`. Das Objekt extrahiert den Testdatensatz, der anschließend angezeigt werden kann.

Erstellen Sie ein Menü, das es dem Bediener ermöglicht, eine Schaltfläche neben der gewünschten Auswahl anzuklicken.

3. Wählen Sie `Data ⇒ Selection Control ⇒ Radio Buttons` aus und platzieren Sie das Objekt links von `For Count`.

   a. Öffnen Sie das Objektmenü und wählen Sie `Edit Enum Values...` aus. Heben Sie `0000: Item 1` hervor und geben Sie `test0` ein. Drücken Sie die Tabulatortaste **Tab**, um zu `0001: Item2` zu wechseln, und geben Sie test1 ein. Wenn Sie nach dem dritten Eintrag (`test2`) die **Tabulatortaste** drücken, erscheint automatisch ein weiterer Eintrag. Fahren Sie mit der Eingabe von Werten fort, bis Sie bei `test9` angelangt sind. Klicken Sie `OK` an, um alle zehn Einträge von `test0` bis `test9` anzuzeigen.

   b. Klicken Sie die Auswahl `Properties` im Objektmenü an, ändern Sie den Objektnamen von `Radio Buttons` in `Test Menu`, wählen Sie `Auto Execute` unter `Execution` aus, wählen Sie `Open View ⇒ Show Terminals` aus und klicken Sie `OK` an.

4. Das Programm kann jetzt immer ausgeführt werden, wenn der Bediener eine Menüauswahl trifft; löschen Sie daher das Objekt `Start`. Klicken Sie mit der rechten Maustaste das Objekt `Start` an und wählen Sie `Cut` aus.

5. Das Programm soll nur ausgeführt werden, wenn eine Menüauswahl getroffen wird. Verbinden Sie daher den Datenausgangs-Pin `Enum` von `Test Menu` mit dem Sequenzeingangs-Pin von `For Count`. Das Programm sollte aussehen wie in Abbildung 5-17.

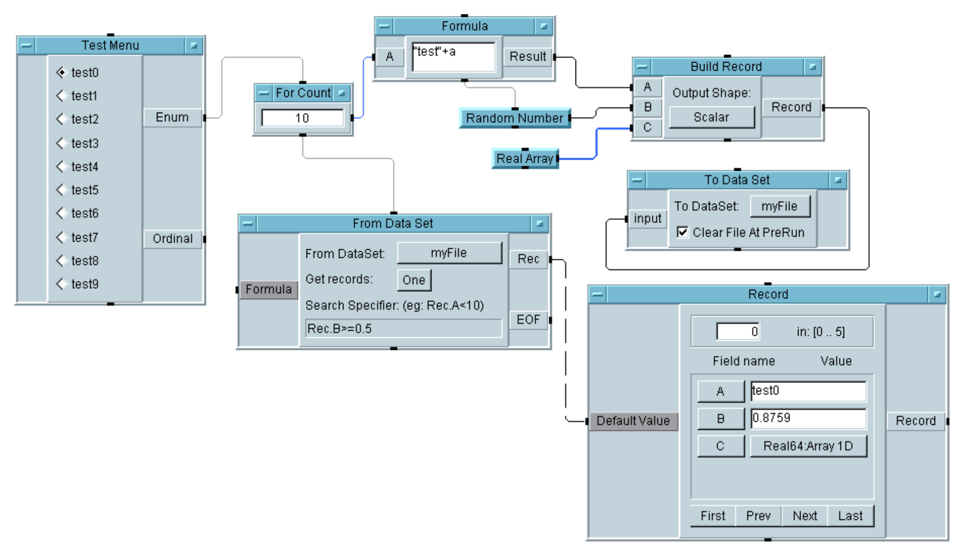


Abbildung 5-17. Hinzufügen des Testmenü-Objekts

6. Der Ausgang des `Test Menu` geht zu einem Objekt `Formula`; dieses wiederum sendet die richtige Formel an das Objekt `From Data Set`.

   Wählen Sie `Device` ⇒ `Formula` aus und platzieren Sie das Objekt unter `Test Menu`. (Sie können beim Hinzufügen von Elementen im Rahmen dieser Übung auch Objekte neu anordnen und/oder ihre Größe ändern.) Geben Sie im neuen `Formula`-Objekt den folgenden Ausdruck ein:

```
"Rec.A==" + "\"" + A + "\""
```

“Rec.A==” — `"Rec.A=="` sendet einen `Text`-Datentyp an den Ausdruckseingang `From Data Formula`. (Die Anführungszeichen kennzeichnen eine Textzeichenfolge.)

A — `VEE` sucht das erste Feld `A` aller Datensätze in der `DataSet`- Datei und wählt den ersten Datensatz aus, der dem ausgewählten Testnamen entspricht.

“\””	Das Escape-Zeichen für Anführungszeichen lautet `\"`. Das Escape-Zeichen wird anschließend in Anführungszeichen gesetzt, um eine Textzeichenfolge zu kennzeichnen. Der *test name* wird vom `Test Menu` als `Enum`-Datentyp empfangen. Anführungszeichen sind erforderlich, um die richtige Formel im Objekt `From DataSet` abzulegen.

Wenn beispielsweise `test6` ausgewählt wird, lautet die endgültige Formel `Rec.A=="test6"`. Das Objekt `From Data Set` gibt dann den ersten gefundenen Datensatz aus, dessen `"A"`-Feld den Wert `"test6"` enthält.

7. Verbinden Sie den Datenausgangs-Pin `Test Menu Enum` mit dem Dateneingangs-Pin am `Formula`-Objekt . Stellen Sie das `Formula`-Objekt als Symbol dar.

8. Verbinden Sie den Datenausgangs-Pin von `Formula` mit dem Steuereingangs-Pin am Objekt `From Data Set` mit der Beschriftung `Formula`.

9. Löschen Sie die Sequenzlinie zwischen `For Count` und `From Data Set`, um sicherzustellen, dass die alten Daten von `Formula` nicht erneut verwendet werden. Verbinden Sie den Sequenzausgangs-Pin des Objekts `For Count` mit dem Sequenzeingangs-Pin von `Formula`.

10. Verbinden Sie den Sequenzausgangs-Pin von `Formula` mit dem Sequenzeingangs-Pin von `From Data Set`. Auf diese Weise wird sichergestellt, dass die richtigen Daten von `Formula` verwendet werden.

11. Erstellen Sie ein Fenster mit Anleitungen für den Bediener. Wählen Sie `Display ⇒ Note Pad` aus. Ändern Sie den Titel in `Test Results Database Instructions`. Klicken Sie den `Note Pad`-Eingabebereich an und geben Sie ein: `Select the test results you want from the Test Menu.`

12. Benennen Sie das Objekt `Record Constant` um in `Test Results`.

13. Das Programm sollte aussehen wie in Abbildung 5-18. Führen Sie das Programm einige Male aus, um sicherzustellen, dass es funktioniert. Da für das Objekt `Test Menu` die Option `AutoExecute` eingeschaltet wurde, erstellen Sie eine Menüauswahl zur Ausführung des Programms.

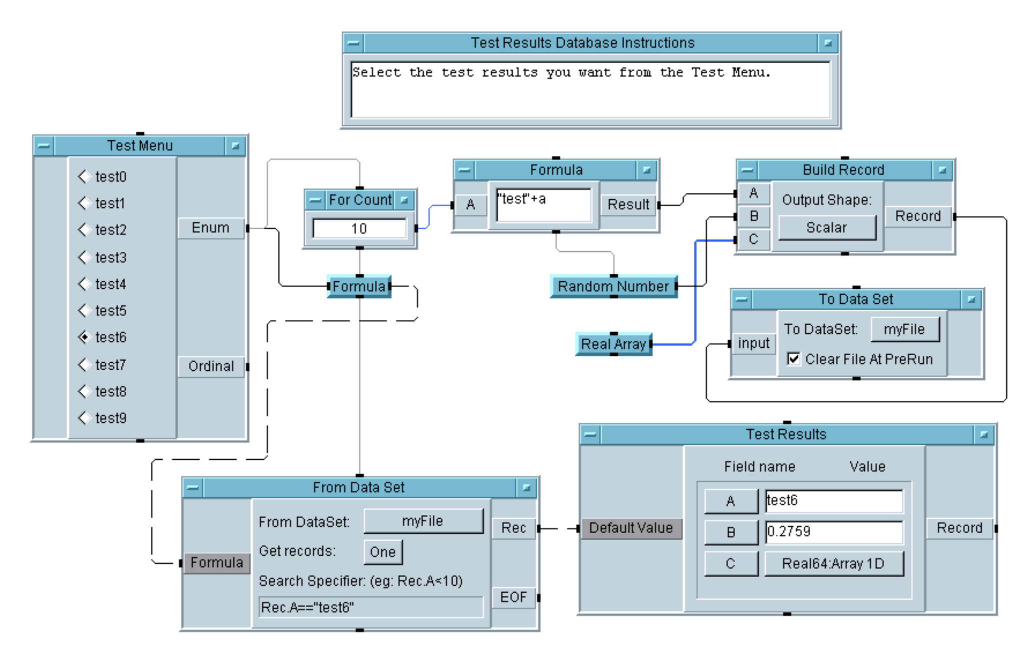


Abbildung 5-18. Hinzufügen eines Menüs zur Suchoperation

Erstellen Sie als nächstes die Bedienerschnittstelle.

14. Drücken Sie **Strg** und klicken Sie die folgenden Objekte an: `Test Menu`, `Test Results Database Instructions` und `Test Results`.

    Alle ausgewählten Objekte zeigen einen Schatten. Vergewissern Sie sich, dass keine weiteren Objekte ausgewählt sind.

    Wählen Sie anschließend `Edit` ⇒ `Add to Panel` aus. Die Bedienerschnittstelle wird als Fensteransicht angezeigt. Sie können jetzt die Objekte verschieben und ihre Größe ändern. Ein Layout ist in Abbildung 5-19 dargestellt.

Hinweis

Wenn die Auswahl `Add to Panel` abgeblendet ist, bedeutet, dies, dass im Arbeitsbereich keine Objekte ausgewählt sind.

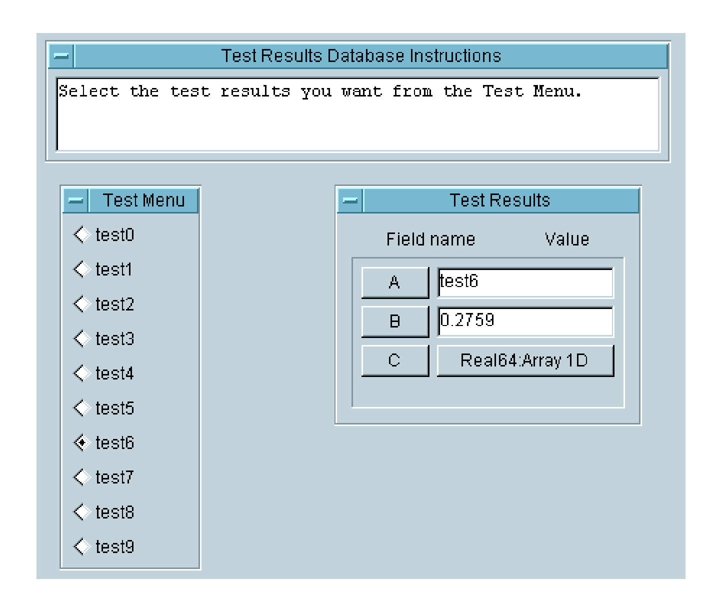


Abbildung 5-19. Die Bedienerschnittstelle für die Datenbank

15. Führen Sie das Programm einige Male aus, indem Sie im `Test Menu` eine Auswahl treffen. Speichern Sie das Programm als `database.vee`.

    Beachten Sie, dass Sie ausführlichere Informationen zu jedem bestimmten Datensatz abrufen können, indem Sie den entsprechenden Feldnamen oder die Werte im Objekt `Record Constant` (mit den Namen `Testergebnisse`) anklicken.

Hinweis

Zum Sichern der Bedienerschnittstelle und des Programms gegen Änderungen wählen Sie `File` ⇒ `Create RunTime Version...` aus. Geben Sie einen Namen für das Programm ein, und VEE fügt automatisch die Erweiterung `*.vxe` hinzu, um das Programm von ungeschützten Versionen zu unterscheiden.

Ausführen einer Sortieroperation mit einem Datensatz-feld

Diese Übung verwendet das Programm `dataset2.vee` aus einer früheren Übung. Das Programm `dataset2.vee` setzt eine Bedingung im Objekt `From DataSet` wie beispielsweise `Rec.B>=0.5`, und VEE extrahiert alle Datensätze, die diese Voraussetzung erfüllen. Der Array der resultierenden Datensätze wird im Objekt `Record Constant` angezeigt.

In dieser Übung wird `dataset2.vee` zur Sortierung der resultierenden Datensätze geändert, um zu ermitteln, welche Tests mit den größten Abweichungen fehlgeschlagen sind. Die Tests sind nach dem zweiten Feld in absteigender Reihenfolge sortiert.

1. Öffnen Sie das Programm `dataset2.vee`.

2. Wählen Sie `Device` ⇒ `Formula` und verbinden Sie den Datenausgangs-Pin `Rec` von `From Data Set` mit dem Dateneingangs-Pin des `Formula`-Objekts. Klicken Sie doppelt auf das Ausdrucksfeld `Formula`, um die Standardformel hervorzuheben, und geben Sie anschließend `sort(a, 1, "B")` ein.

   Das Objekt `Sort` befindet sich in den Funktionen `Function & Object Browser`, `Array Category`. Sie können ausführliche Informationen zu seinen Funktionen im Objektmenü unter `Help` nachschlagen. Die Funktion `sort ()` wird vom `Formula`-Objekt aufgerufen.

   Der erste Parameter sortiert die Daten am Pin `A` des `Formula`-Objekts in einem Array von Datensätzen. Der zweite Parameter kennzeichnet die Richtung der Sortierung: Jede Zahl ungleich Null kennzeichnet eine aufsteigende Sortierfolge, Null kennzeichnet eine absteigende Sortierung. Die Standardrichtung ist aufsteigend. Der dritte Parameter bei einem Datentyp `Record` kennzeichnet den Namen des Felds, nach dem sortiert werden soll. In diesem Beispiel wird also eine aufsteigende Sortierung nach dem Feld `B` in dem Arrays der Datensätze durchgeführt.

3. Wählen Sie `Display` ⇒ `AlphaNumeric` und verbinden Sie das Objekt mit dem Datenausgangs-Pin des `Formula`-Objekts.

4. Führen Sie das Programm mehrmals aus. Es sollte jetzt aussehen wie in Abbildung 5-20. Beachten Sie, dass das Programm alle von der Datei `DataSet` zurückgegebenen Datensätze in aufsteigender Reihenfolge nach dem Feld `B` sortiert.

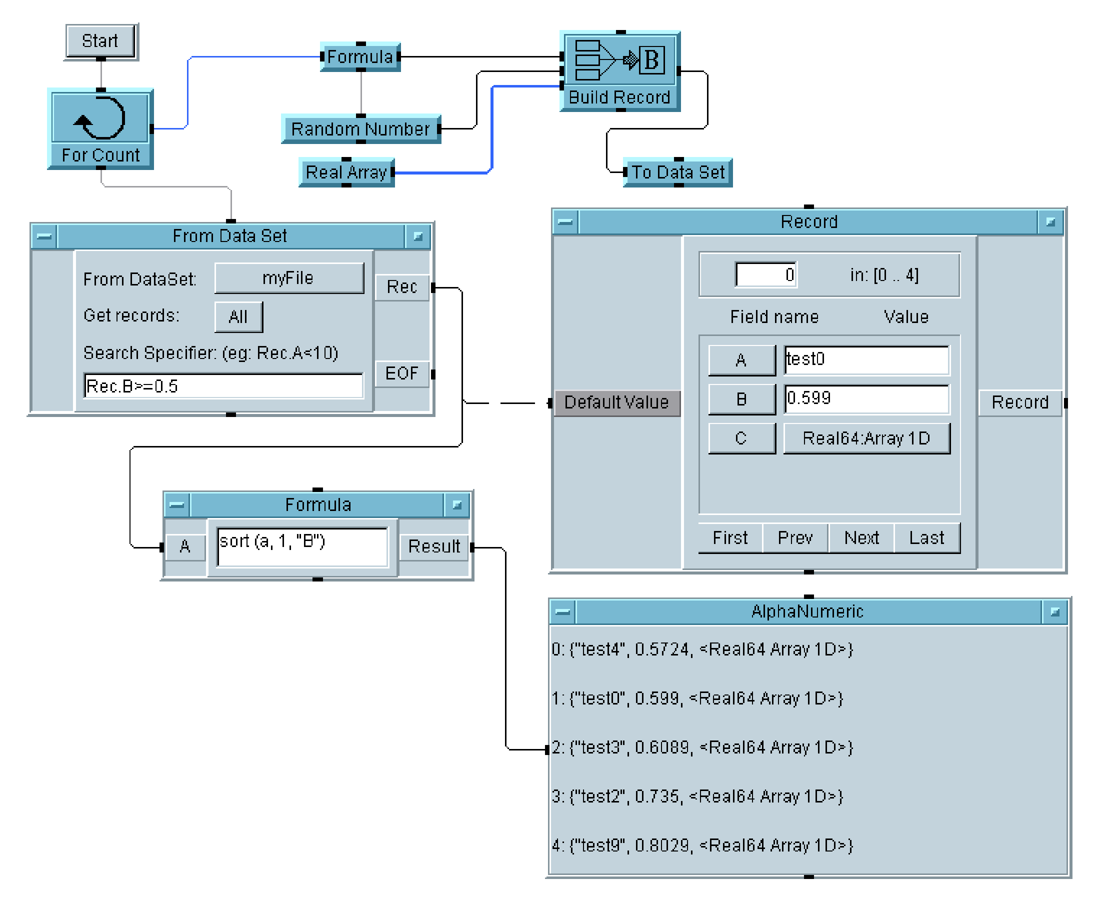


Abbildung 5-20. Eine Sortieroperation mit einem Datensatzfeld

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die Themen ggf. noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Erläutern der Basisnotation zur Verwendung von Arrays.
- Erstellen eines Array mit dem Objekt `Collector`.
- Extrahieren von Elementen aus einem Array mit dem `Formula`-Objekt.
- Senden einer Zeichenfolge, eines Zeitstempels und eines Real-Array an eine Datei.
- Abrufen einer Zeichenfolge, eines Zeitstempels und eines Real-Array aus einer Datei.
- Verwenden der Funktion `now()` für einen Zeitstempel.
- Formatieren der Zeitstempel auf verschiedene Arten für die Anzeige.
- Erstellen und Auflösen eines Datensatzes.
- Abrufen und Einstellen von Feldern in einem Datensatz.
- Speichern eines Datensatzes in einem `DataSet`.
- Abrufen eines Datensatzes aus einem `DataSet`.
- Durchführen einer Suchoperation mit einem `DataSet`.
- Durchführen einer Sortieroperation mit einem `Record`-Feld.
- Kombinieren von VEE-Tools zum Erstellen einer einfachen Datenbank.

6

Einfaches Erstellen von Berichten mit ActiveX

Einfaches Erstellen von Berichten mit ActiveX

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- ActiveX-Automatisierung in VEE
- Verwenden von ActiveX für Berichte mit MS Excel
- Verwenden von ActiveX für Berichte mit MS Word

Erforderliche Zeit für dieses Kapitel: 1,5 Stunden

Überblick

In diesem Kapitel lernen Sie, wie Sie Berichte in anderen Anwendungen wie beispielsweise MS Excel generieren, indem Sie Daten von dem VEE-Programm an das MS Excel-Programm senden. VEE verwendet die ActiveX-Automatisierung zur Steuerung anderer Anwendungen; dies ermöglicht ein schnelles Erstellen ausführlicher und effektiver Berichte.

In der ersten Übung wird beschrieben, wie Sie Daten an eine MS Excel-Tabelle senden, die automatisch die ActiveX-Automatisierung verwendet. Die zweite Übung beschreibt allgemeine Vorlagen zum Generieren von Berichten und das Erweitern der Funktionalität der Basisvorlage. Die letzte Übung verwendet ActiveX in VEE zum Senden eines Bildschirmabzugs und der Testdaten an ein MS Word-Dokument. (Das Prinzip ist für andere Tabellenkalkulationen und Textverarbeitungsprogramme mit Unterstützung der ActiveX-Automatisierung das gleiche.)

Hinweis

ActiveX ersetzt die Verwendung von DDE in VEE. DDE wird jedoch in VEE weiterhin unterstützt. Informationen zur Verwendung von DDE in herstellerspezifischen Anwendungen finden Sie in der zweiten Ausgabe von *Visual Programming with HP VEE*.

ActiveX-Automatisierung in Agilent VEE

In diesem Kapitel bezeichnet der Begriff ActiveX-Automatisierung die Möglichkeit von VEE, als ein Automation Controller von Automatisierungs-Server-Anwendungen wie MS Excel, MS Word oder MS Access zu arbeiten. Der Schwerpunkt dieser Übung liegt auf der praktischen Anwendung der Microsoft ActiveX-Technologie zum Generieren von Programmberichten aus dem Bereich Test- und Messprogramme.

Hinweis

Dieses Handbuch enthält auch weitere entsprechende Übungen in den Abschnitten “Verwenden eines ActiveX-Steuerelements” auf Seite 417 und “Der Aufrufbare VEE ActiveX-Automatisierungs-Server” auf Seite 471. Weitere ausführliche Informationen zum Konzept und der Terminologie der Automatisierung finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Auflisten von Typenbibliotheken zur ActiveX-Automatisierung

Zum Auflisten der Automatisierungssobjekte auf Ihrem Computer klicken Sie `Devices ⇒ ActiveX Automation References` an.

Hinweis

Informationen zu ActiveX Steuer-Referenzen finden Sie in Kapitel 10, “Verwenden von Bedienerschnittstellen”. Schlagen Sie außerdem nach in Kapitel 12, “Plattformspezifische Faktoren und Web-Überwachung”.

`Devices ⇒ ActiveX Automation References` listet die auf Ihrem PC installierten Typenbibliotheken auf. Jede Anwendung und ActiveX-Komponente, die als Automatisierungs-Server eingesetzt werden kann, registriert eine Typenbibliothek. VEE zeigt an, was auf Ihrem PC verfügbar ist. Diese Bibliotheken enthalten Informationen zur Funktionalität der Anwendung oder Komponente, die den ActiveX-Clients bereitgestellt wird.

Typenbibliotheken bestehen normalerweise aus einer Gruppe von Klassen. Manche Klassen können vom Programmierer erstellt werden. Andere Klassen werden immer von der Anwendung oder Komponente erstellt. Klassen umfassen Eigenschaften, Methoden und Ereignisse; nicht alle diese Kompo-

nenten müssen vorhanden sein. Die Typenbibliothek bietet dem Programmierer und der VEE-Umgebung die erforderlichen Informationen zur Nutzung der Anwendung oder Komponente mit ActiveX-Schnittstellen.

Wenn Sie ein Häkchen neben einer Typenbibliothek im Feld ActiveX Automation References eintragen, sind die Bibliotheksobjekte zur Verwendung in einem VEE-Programm verfügbar. In Abbildung 6-1 ist beispielsweise Microsoft Excel 9.0 markiert.

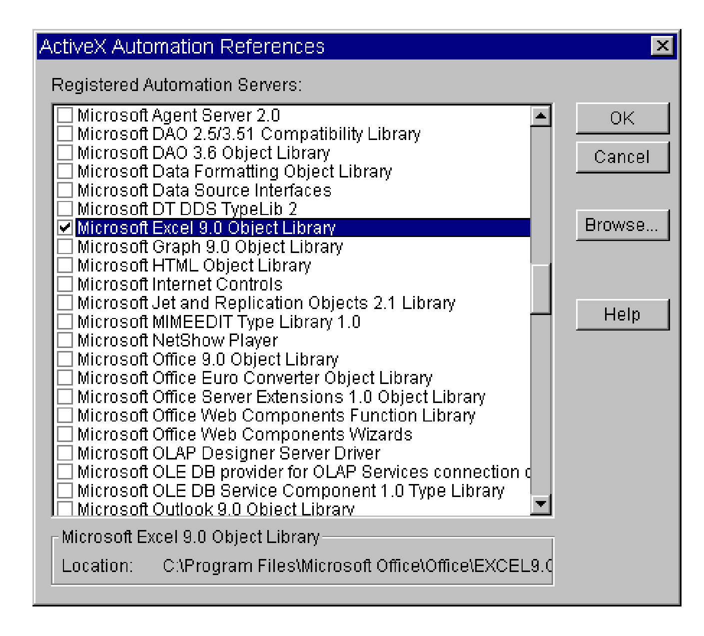


Abbildung 6-1. Das Fenster "ActiveX Automation References"

Erstellen und Verwenden von ActiveX-Programmen mit Agilent VEE

VEE umfasst den Datentyp `Object` für ActiveX-Programme. Ein VEE-Objekt mit dem Datentyp `Object` ist ein Zeiger auf eine Komponente oder ein Datenelement, die bzw. das auf dem Automatisierungs-Server vorhanden ist. Ein `Object` kann beispielsweise auf ein Arbeitsblatt in MS Excel verweisen oder auf eine Zelle in diesem Arbeitsblatt. (Technisch gesehen ist ein `Object` ein Zeiger auf eine IDispatch-Schnittstelle, der von MS Excel oder dem Server zurückgegeben wird.)

Wenn Sie beispielsweise `Data ⇒ Variable ⇒ Declare Variable` auswählen, `Name` auf `App` festlegen und den Datentyp als `Object` definieren, können Sie über die Variable `App` auf ein ActiveX-Automatisierungsobjekt wie beispielsweise den Excel Automatisierungs-Server verweisen. Abbildung 6-2 zeigt ein Beispiel des Datentyps `Object`.

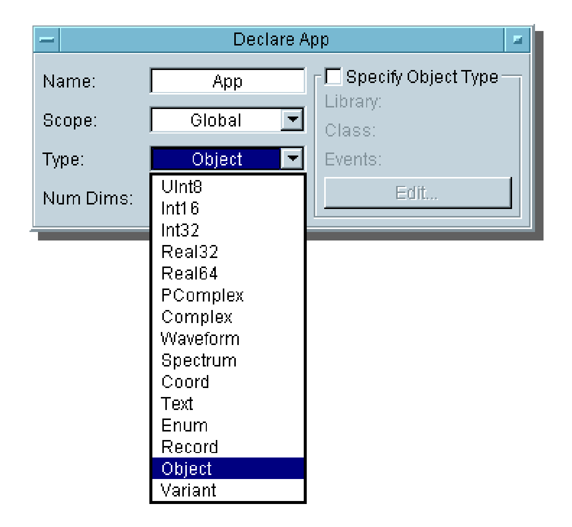


Abbildung 6-2. Beispiel des Datentyps “Object”

Durchführen von Operationen mit ActiveX-Anweisungen

Zur Kommunikation mit einem ActiveX Automatisierungs-Server wie beispielsweise dem Excel Automatisierungs-Server, geben Sie ActiveX-Befehle in einem VEE `Formula`-Objekt ein. Abbildung 6-3 zeigt beispielsweise ein VEE `Formula`-Objekt mit dem Namen `Set Up Excel Worksheet`. Es enthält eine Liste von Befehlen zum Einrichten eines Excel-Arbeitsblatts zur Anzeige der Ergebnisse eines Tests.

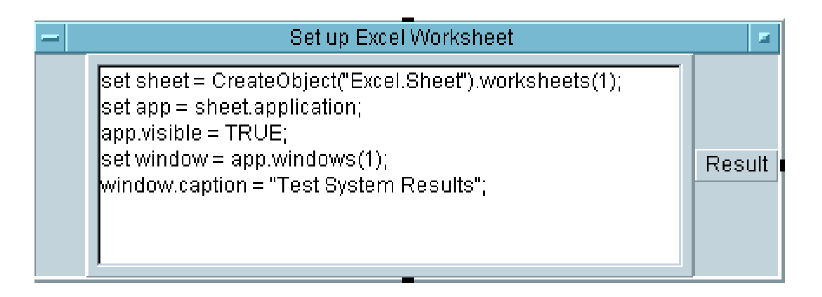


Abbildung 6-3. Befehle zum Einrichten eines Excel-Arbeitsblatts zur Anzeige der Testergebnisse

VEE verwendet die Standardsyntax von Microsoft Visual Basic zum Erstellen von Befehlen oder Anweisungen wie die in Abbildung 6-3 gezeigten. Die Befehle oder Anwendungen führen drei Arten von Operationen aus: *get properties* (Eigenschaften abrufen), *set properties* (Eigenschaften festlegen) oder *call methods* (Methoden aufrufen).

- *Get property*-Anweisungen beziehen sich normalerweise auf das Abrufen von Daten. Die Syntax lautet *<Objekt>*`.`*<Eigenschaft>*. `sheet.application` ruft beispielsweise die Eigenschaft `application` des Objekts `sheet` ab.

- *Set property*-Anweisungen beziehen sich normalerweise auf das Setzen von Daten auf einen bestimmten Wert. Die Syntax lautet *<Objekt>*`.`*<Eigenschaft>* `=` *<Eigenschaftstyp>*. `object.property = MaxSize` legt beispielsweise eine Eigenschaft fest.

- *Call methods* rufen eine `Methode` auf. Eine `Methode` fordert von dem Objekt die Ausführung einer Aktion an. `Methoden` enthalten Parameter, die das Übergeben und Zurückgeben von Parametern ermöglichen. Die Syntax lautet *<Objekt>*`.`*<Methode>(Parameter)*.

Hinweis

Die Syntax für den Datentyp `Objects` ähnelt der VEE-Syntax zum Abrufen eines Record-Felds `rec.field` und dem Aufruf einer `UserFunction`, `myLib.func()`. Es ist daher wichtig, dass Sie den Variablen beschreibende Namen zuordnen.

Verwenden von CreateObject und GetObject

Beachten Sie, dass eine der Anweisungen in `Set Up Excel Worksheet` in Abbildung 6-3 den Funktionsaufruf `CreateObject()` enthält. `CreateObject()` und `GetObject()` sind Funktionen im VEE `Function & Object Browser`; sie wurden speziell für die Rückgabe eines Zeigers an ein ActiveX-Objekt in VEE konzipiert.

`CreateObject("Excel.Sheet")` startet beispielsweise Excel und gibt eine Referenz auf eine darin enthaltene Arbeitsmappe zurück. (Die Microsoft Anweisung “sheet” gibt eine Arbeitsmappe zurück.) Verwenden Sie `GetObject()` zum Abrufen von bereits vorhandenen Daten aus einer aktiven Excel-Tabelle oder zum Laden einer Datei in ein aktives Excel.

`CreateObject` und `GetObject` befinden sich unter `Device ⇒ Function & Object Browser`, `Type: Built-in Functions`, `Category: ActiveX Automation`. Abbildung 6-4 zeigt ein Beispiel zu `CreateObject` und `GetObject`.

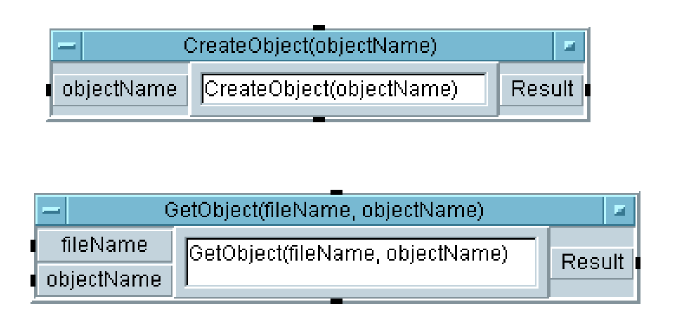


Abbildung 6-4. CreateObject und GetObject

Senden von Agilent VEE-Daten an MS Excel

In diesem Abschnitt werden die VEE-Objekte und MS Excel-Funktionsaufrufe zum Generieren von Berichten vorgestellt.

Übung 6-1: Senden von Agilent VEE-Daten an MS Excel

In dieser Übung generieren Sie virtuelle Testdaten für MS Excel. (Das Beispiel verwendet MS Office 2000 und die MS Excel 9.0 Objektbibliothek; es sollte jedoch auch mit MS Office 97 und der MS Excel 8.0 Objektbibliothek verwendet werden können.) Nach dem Verweisen auf die richtige Automatisierungs-Typenbibliothek deklarieren Sie einige globale Variablen des Typs `Object` und legen sie in einer `UserFunction` mit dem Namen `globals` ab. Die globalen Variablen vereinfachen das Programm und machen es leichter verständlich.

Hinweis

Die VEE -Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter `Help` ⇒ `Open Example...` ⇒ `Manual` ⇒ `UsersGuide` enthalten.

1. Verweisen Sie auf die Automatisierungs-Bibliothek. Klicken Sie `Device` ⇒ `ActiveX Automation References...` an, wählen Sie `Microsoft Excel 9.0 Object Library` aus und klicken Sie `OK` an.

2. Erstellen Sie eine `UserFunction` zum Speichern der globalen Variablen. Klicken Sie `Device` ⇒ `UserFunction` an. Benennen Sie diese UserFunction um in `globals`. (Weitere Informationen zu `UserFunctions` finden Sie in Kapitel 8, “Verwenden von Agilent VEE-Funktionen” auf Seite 309.)

3. Klicken Sie `Data` ⇒ `Variable` ⇒ `Declare Variable` an und platzieren Sie dieses Objekt auf der linken Seite innerhalb von `globals`. Ändern Sie die Angabe `Name` in `sheet`. Ändern Sie die Angabe `Type` in `Object`. In dem Dialogfenster werden weitere Elemente angezeigt. Für diese Übung brauchen Sie `Object Type` und `Class` nicht anzugeben.

(`Type` und `Class` werden in einem anderen Beispiel in diesem Kapitel angegeben.)

4. Klonen Sie dieses Objekt drei Mal, und benennen Sie die anderen Objekte wie folgt um: `app`, `range` und `window`. Ändern Sie die Größe und verschieben Sie die `globals UserFunction` unter `Main`. Es sollte aussehen wie in Abbildung 6-5.

5. Vergleichen Sie die Einträge mit der Anzeige in Abbildung 6-5, und stellen Sie die vier Objekte anschließend als Symbole dar.

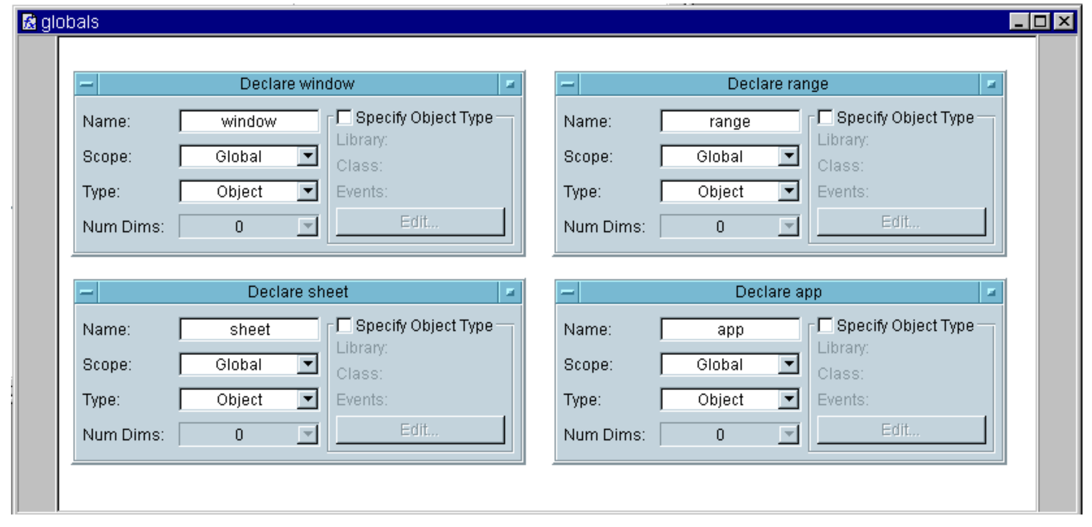


Abbildung 6-5. Die UserFunction "Globals"

Beachten Sie, dass Sie durch die Verwendung des Datentyps `Objects` bei der `UserFunction globals` auch `Object Type und Class` angeben können. Es gibt zwei Gründe, `Object Type` und `Class` anzugeben: ausführliche Typenprüfung und das Erfassen von Ereignissen.

Ausführliche Typenprüfung: Wenn Sie beispielsweise für eine `Object app` den Typ `Excel.Application` festlegen, kann nur ein `Object` des Typs `Excel.Application` der `app` zugeordnet werden. Das Zuordnen

eines `Object` des Typs `Excel.worksheet` oder `Word.bookmark` führt zu einem Fehler.

Erfassen von Ereignissen: Sie können eine VEE `UserFunction` auch verwenden, um verschiedene *Ereignisse* zu erfassen, die in der Anwendung auftreten können, beispielsweise das Drücken der rechten Maustaste in einer MS Excel-Tabelle. Für alle diese Ereignistypen können Sie eine VEE `UserFunction` angeben, die das Ereignis verarbeitet und Informationen an MS Excel zurückgibt. Ereignisse sind hilfreich für ActiveX-Steuerelemente, bei denen Sie eine Möglichkeit brauchen, die Kommunikation zurück zu VEE zu steuern. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

6. Öffnen Sie das Objektmenü der `UserFunction` `globals` und klicken Sie `Generate` ⇒ `Call` an. Dadurch wird ein korrekt konfiguriertes Objekt `Call globals` generiert. Platzieren Sie dieses Objekt links im `Hauptfenster` und stellen Sie das Fenster `globals UserFunction` als Symbol dar.

7. Klicken Sie `Device` ⇒ `Formula` an und platzieren Sie dieses Objekt oben in der Mitte des Hauptfensters. Benennen Sie es um in `Set up Excel Worksheet`. Verbinden Sie den Sequenzausgangs-Pin des Objekts `globals` mit dem Sequenzeingangs-Pin von `Formula`. Löschen Sie den Eingangsanschluss `A` von `Set Up Excel Worksheet` (öffnen Sie das `Objektmenü` und wählen Sie `Delete Terminal` ⇒ `Input` aus.)

8. Geben Sie in `Set up Excel Worksheet` die Zeilen wie in Abbildung 6-6 gezeigt ein. Beachten Sie, dass für die Abgrenzung der Zeilen Semikolons verwendet werden wie in ANSI C.

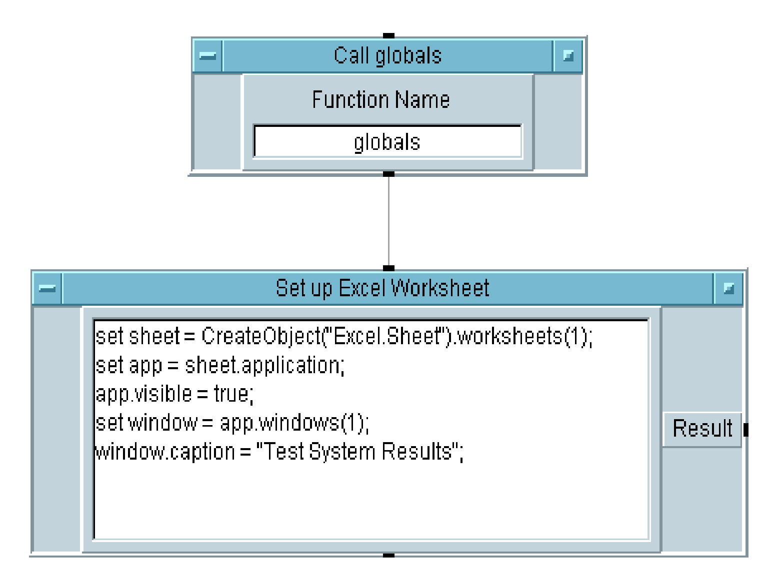


Abbildung 6-6. Einrichten der MS Excel-Tabelle

Die Befehle im `Formula`-Objekt `Set Up Excel Worksheet` lauten wie folgt:

set sheet = Mit dem Schlüsselwort `set` wird der Variablen auf der linken Seite des Ausdrucks die Angabe rechts vom Zuordnungsoperator (in diesem Fall das Gleichheitszeichen) zugeordnet. `set app` definiert beispielsweise `sheet.application`, die als Excel-Tabelle definiert wurde.

CreateObject ("Excel.Sheet").

Erstellt ein neues Exemplar des Automatisierungs-Servers (in diesem Fall MS Excel) und erstellt ein leeres Blatt (Excel-Terminologie für eine neue Arbeitsmappe). Jeder Automatisierungs-Server hat seine eigene Terminologie; die Syntax ist jedoch für alle gleich. Zum Erstellen eines Berichts in MS Word geben Sie beispielsweise `CreateObject ("Word.Document")` ein, um Word zu starten und ein leeres Dokument zu erstellen.

Bei Verwendung des Schlüsselworts `set` wird die rechte Seite des Objektzeigers der Variablen auf der linken Seite zugeordnet. Wenn `set` nicht angegeben ist, wird die Standardeigenschaft (häufig der Name) von der rechten Seite der linken Seite zugeordnet. Weitere Informationen hierzu finden Sie im Handbuch *VEE Pro Advanced Techniques*.

worksheets(1);

Excel wird jetzt mit einer neuen Arbeitsmappe ausgeführt. Mit `CreateObject("Excel.Sheet")` können Sie die erste Arbeitsmappe in Excel adressieren. Fügen Sie `worksheets(1)` der Anweisung hinzu, sodass die gesamte Anweisung lautet:

```
setsheet =
CreateObject("Excel.Sheet").worksheets(1);
```

Dadurch wird `sheet` auf `Sheet 1` des Berichts festgelegt. (Sie können hierzu ein Beispiel anzeigen, indem Sie MS Excel öffnen und `Datei` ⇒ `Neu` auswählen, um eine neue Arbeitsmappe zu erstellen. Sie werden feststellen, dass darin mehrere Blätter mit den Namen `Sheet1`, `Sheet2` etc. vorhanden sind. Sie wollen `Sheet1` verwenden.)

set app = sheet.application;

Ruft aus Excel einen Zeiger auf die gesamte Anwendung ab und nicht nur auf die aktuelle Arbeitsmappe. Hierzu wird die Eigenschaft `Application` der Tabelle abgefragt und auf die Variable `app` gesetzt.

app.visible = true;

Setzt die Sichtbarkeitseigenschaft `app` auf den Wert `true`, um Excel auf dem Bildschirm anzuzeigen (sodass es für Sie sichtbar wird).

set window = app.windows(1);	Verweist auf das erste Fenster.
window.caption = “Test System Results”;	Setzt die Legende für das erste Fenster auf “Test System Results.”

Hinweis

Weitere Informationen zu den Bibliotheken des Anwendungs-Servers finden Sie in einer Vielzahl von Büchern über die ActiveX-Automatisierung und MS Visual Basic. Informationen zur Bestellung von Büchern wie dem Office 2000 oder Office 97 Visual Basic Programmer’s Guide finden Sie unter anderem im World Wide Web. Diese Bücher helfen Ihnen auch bei VEE weiter, da die VEE-Syntax der Syntax von MS Visual Basic stark ähnelt.

9. Erstellen Sie ein `Formula`-Objekt (unter `Device ⇒ Formula)`. Klonen Sie das `Formula`-Objekt, um ein zweites `Formula`-Objekt zu erstellen. Erstellen Sie das Objekt `For Range` (unter `Flow ⇒ Repeat ⇒ For Range`). Benennen Sie die Objekte um, verbinden Sie sie und konfigurieren Sie sie wie in Abbildung 6-7 gezeigt. (Vergewissern Sie sich, dass Sie den Eingangsanschluss am `Formula`-Objekt `Fill in Title` gelöscht haben.)

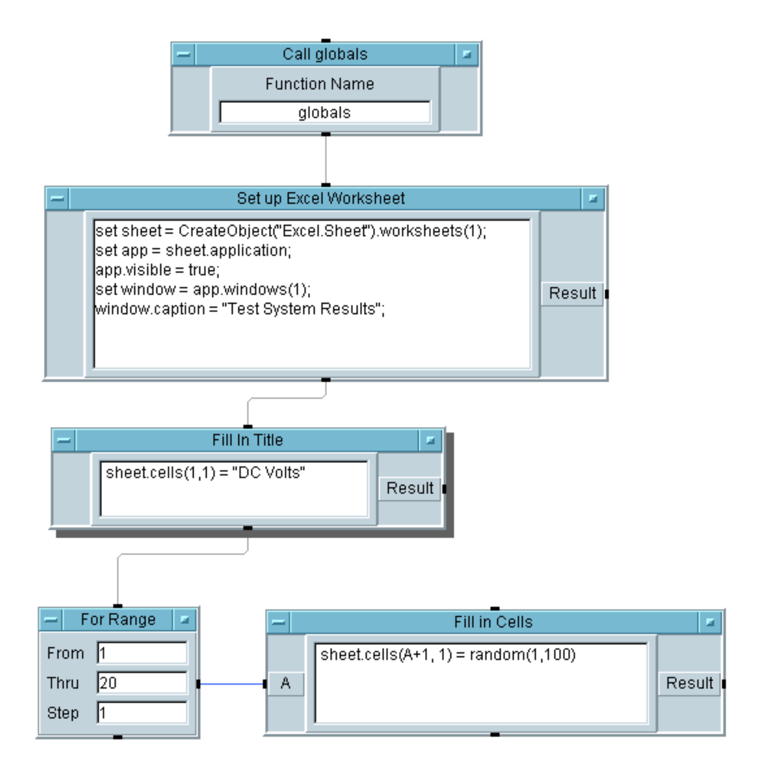


Abbildung 6-7. Hinzufügen von Titel und Daten zu dem Blatt

Die Anleitungen in den `Formula`-Objekten und dem Objekt `For Range` lauten wie folgt:

sheet.cells(1,1) = “DC Volts” Bezieht sich auf die erste Zeile und Spalte in der Excel-Tabelle. Der Text "DC Volts" wird hier platziert. Setzt die Standardeigenschaft (den Wert) der Zelle `(1,1)` auf `"DC Volts"`.

sheet.cells(A+1,1) = random(1,100)	Diese Anweisung ist eine Kurzschrift für `sheet.cells(A+1,1).value=random(1,100)`. Die Tabellenzelle in Zeile `A+1, col 1` ruft die Zeilennummer ab durch Addieren von `1` zum Wert des Eingangs-Pin `A`; Spalte `1` wird jedoch beibehalten. Der von `random` zurückgegebene Wert zwischen `1` und `100` wird der angegebenen Zelle in der Tabelle zugeordnet.
von 1 bis 20, Schritt 1 (das Objekt "For Range")	Da das Objekt `For Range` die Ganzzahlen von `1` bis `20` ausgibt, legt `Fill in Cells` die Zufallszahl in der angegebenen Zelle ab.

10. Erstellen Sie ein `Formula`-Objekt und ein `AlphaNumeric`-Objekt. Benennen Sie diese Objekte anschließend um, konfigurieren und verbinden Sie sie, wie in Abbildung 6-8 gezeigt.

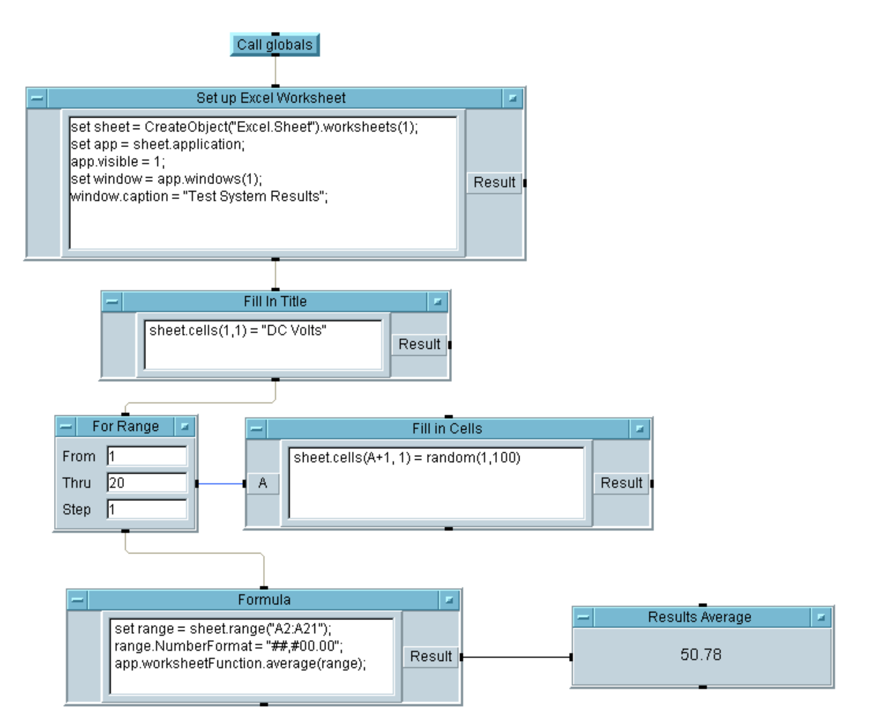


Abbildung 6-8. Das Programm "Results Average"

Die Einträge im `Formula`-Objekt lauten wie folgt:

set range = sheet.range(“A2:A21");	Setzt den VEE-Variablenbereich als Verweis auf den Bereich `A2` bis `A21` in der Excel-Tabelle. (`A` bezieht sich auf die erste Spalte in einer Tabelle.)
range.NumberFormat = “##,#00.00";	Ordnet das Format jeder der Zellen mit einem Pfundzeichen (`#`) zu; dies ermöglicht bei Bedarf größere Zahlen.
app.worksheetFunction.average (range);	Ruft die Excel-Methode `average()` auf, die den Durchschnittswert aus dem angegebenen `Bereich` von Werten angibt; dieser Durchschnittswert wird in `Results Average` angezeigt.

11. Speichern Sie das Programm als `results_average.vee`. Führen Sie das Programm aus. MS Excel wird mit einer Tabelle wie der in Abbildung 6-9 gezeigten aufgerufen.

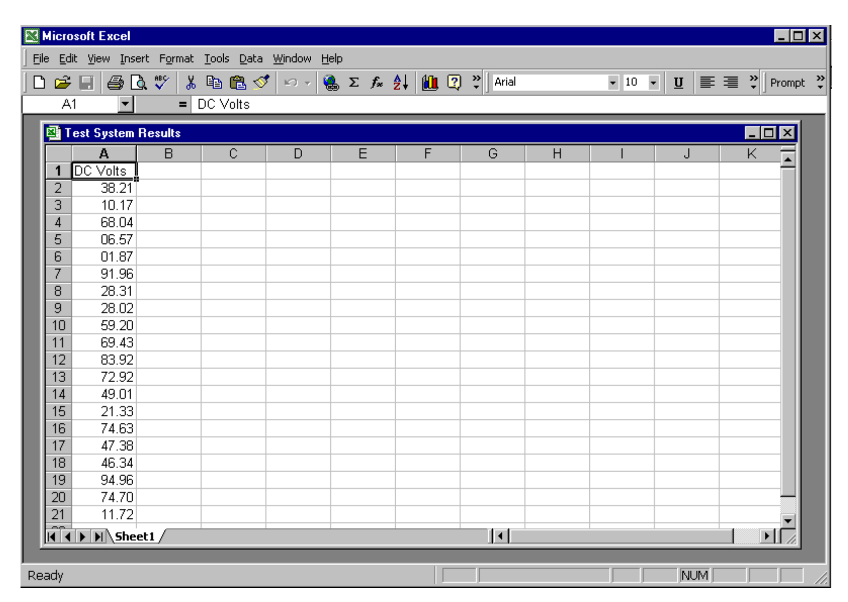


Abbildung 6-9. Excel-Tabelle für das Programm “Results Average”

Erstellen einer Agilent VEE-Vorlage für MS Excel

In dieser Übung erstellen Sie ein Programm zum Anzeigen eines Array von VEE-Testdaten in MS Excel. Sie können dieses Programm als Vorlage zum Anzeigen der Ergebnisse aus anderen Tests in MS Excel-Tabellen verwenden.

Übung 6-2: Erstellen einer Agilent VEE-Vorlage für MS Excel

1. Öffnen Sie `results_average.vee`.

2. Ändern Sie das Objekt `For Range`, sodass es `10` Mal eine Schleife durchläuft.

3. Addieren Sie den Eingang `B` zu dem Objekt `Fill in Cells`, und ändern Sie die darin enthaltene Anweisung wie folgt: `sheet.cells(A+1,1) = B[A-1]`.

   Klicken Sie `Device` ⇒ `Formula` an, benennen Sie das Objekt um in `Array of Test Data` und geben Sie die integrierten Funktionen `randomize(ramp(20), 4.5, 5.5)` ein, um einen Zufalls-Array aus `20` Elementen mit Werten von `4.5` bis `5.5` zu generieren. Löschen Sie den Eingangs-Pin und verbinden Sie den Datenausgangs-Pin mit dem Eingang `B` von `Fill in Cells`.

4. Ändern Sie den Bereich im Fenster `Formula` am unteren Rand der Anzeige von `A21` in `A11`. Die Anweisung sollte jetzt lauten: `set range = sheet.range("A2:A11");`

5. Speichern Sie das Programm als `report_template.vee` und führen Sie es aus. Vergleichen Sie es mit der Excel-Tabelle in Abbildung 6-10 und dem fertigen Programm in Abbildung 6-11.

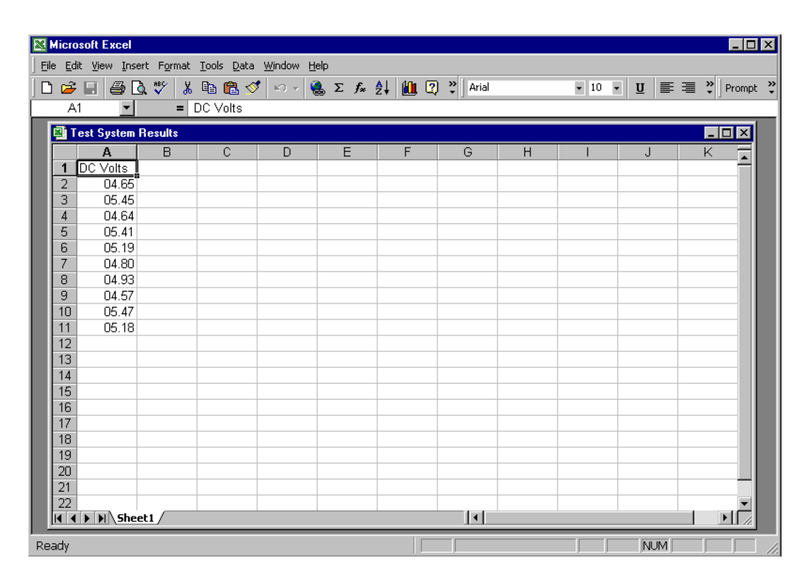


Abbildung 6-10. Excel-Tabelle aus dem Array von Testdaten

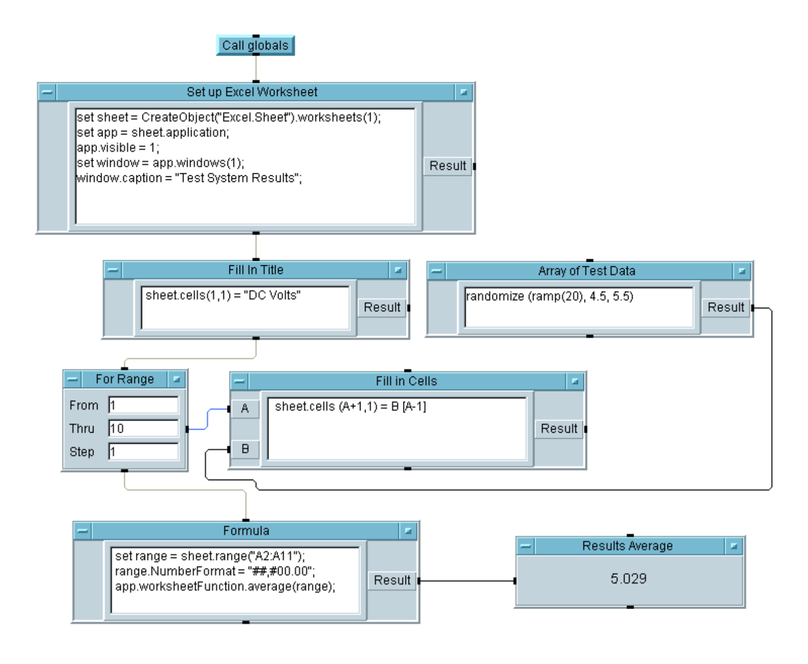


Abbildung 6-11. Programm für einen Array von Testdaten

Sie können dieses Programm als Vorlage zum Anzeigen von Testergebnissen in MS Excel verwenden. Legen Sie einfach die Testdaten in Arrays ab und ändern Sie den Rest der Vorlage so, dass die entsprechenden Zellen im richtigen Format ausgefüllt werden.

Zusätzliche Methoden und Eigenschaften in der MS Excel-Bibliothek können Sie über den `Function & Object Browser` anzeigen; wählen Sie `ActiveX Objects` unter `Type` und `Excel` unter `Library` aus. Sie können eine `Class` oder ein `Member` auswählen und `Help` anklicken, um Hilfetexte zum Automatisierungs-Server vom jeweiligen Hersteller (in diesem Fall Microsoft) aufrufen. Weitere ausführliche Informationen zu diesen Bibliotheken finden Sie in der Microsoft Dokumentation.

Eigene Experimente

Generieren Sie eine Wellenform und entfernen Sie die Angabe `Time Span`, um einen Array zu erhalten. Erstellen Sie ein VEE-Objekt für MS Excel mit einer Tabelle, und stellen Sie es auf eine `Object`-Variable ein. Machen Sie die Anwendung sichtbar. Legen Sie anschließend den 256-Punkt-Array im Tabellenbereich `"A1:A256"` in einem Arbeitsschritt ab statt Zelle für Zelle.

TIPPS: Verwenden Sie ein Objekt `Unbuild Waveform`. Verwenden Sie die Build-Array-Syntax `[a]` zum Erstellen eines 2D-Array aus einem 1D-Array. Rufen Sie anschließend die Funktion `transpose()` auf, um einen 256 x 1-Array zu generieren statt eines 1 x 256-Array. Somit kann Excel den Array in einem Schritt akzeptieren (siehe Abbildung 6-12).

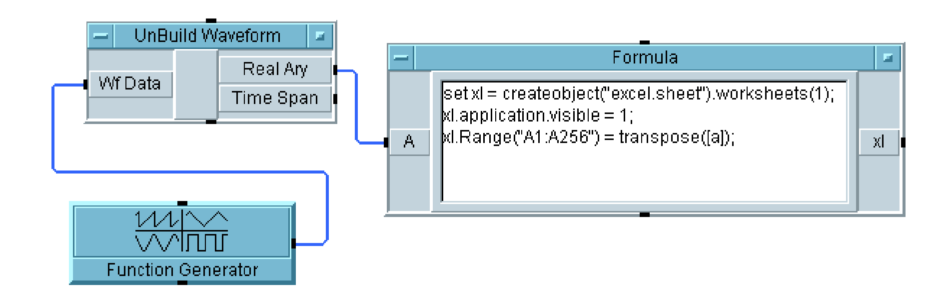


Abbildung 6-12. Programm zur eigenen Übung

Erweitern der Möglichkeiten mit MS Excel

Abbildung 6-13 zeigt ein besser ausgearbeitetes Beispiel eines Programms zum Anzeigen der Testergebnisse in MS Excel. Sie sehen, wie die Informationen über einige weitere Aufrufe in der MS Excel-Bibliothek die Vorlage zum Anzeigen der VEE-Daten in MS Excel erweitern.

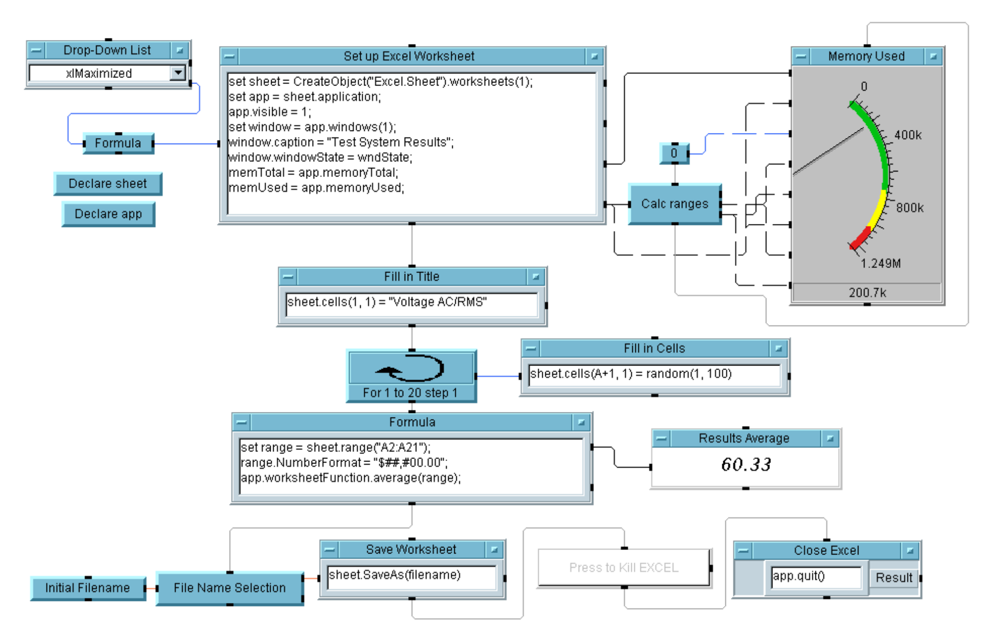


Abbildung 6-13. Ein Beispielprogramm VEE zu MS Excel

Die Einträge in Abbildung 6-13 lauten wie folgt:

MS Excel Window Size

Beachten Sie die Dropdown-Liste oben rechts im Arbeitsbereich. Diese Liste ermöglicht die Auswahl einer von drei möglichen Optionen `xlMaximized`, `xlMinimized`, `xlNormal` für die Größe des Tabellenfensters beim Aufruf in Excel. Jeder Fenstergröße ist eine Zahl zugeordnet, die VEE berechnet und in der Variablen `wndState` ablegt. Dieser Wert wird dann der Eigenschaft `windowState` in der Excel-Bibliothek zugeordnet.

Memory Tracking	(Klicken Sie `Show Terminals` in den Feldern `Properties` der Objekte `Formula` und `Meter` an.) Beachten Sie die Eigenschaften `memoryTotal` und `memoryUsed` in der Excel-Bibliothek, die den VEE- Variablen `memTotal` und `memUsed` zugeordnet sind. Diese Werte werden zum Berechnen der Bereiche für die Konfiguration einer VEE-Messanzeige verwendet, bevor diese die von MS Excel verwendete Speichermenge anzeigt.
Number Format	Beachten Sie, wie einfach es ist, dem Zahlenformat ein Dollarzeichen hinzuzufügen.
sheet.SaveAs (Dateiname)	Die Methode `SaveAs()` wird von der Excel-Bibliothek zum automatischen Speichern der Tabelle aufgerufen. Beachten Sie, dass ein Fenster `File Name Selection` (über das Menü `Data` ⇒ `Dialog Box`) das Einblendfenster `Save As` von VEE angezeigt wird. Der ausgewählte Dateiname wird anschließend als Parameter im Aufruf der Methode `Excel SaveAs()` verwendet.
Press to Kill Excel	Die Bestätigungsschaltfläche (`OK`) wurde verwendet, um das Schließen von Excel zu bestätigen.
Close Excel	Die Methode `quit()` wird aufgerufen, um MS Excel zu verlassen.

Verwenden von MS Word für Agilent VEE-Berichte

In dieser Übung wird beschrieben, wie VEE-Testinformationen einschließlich Text, Zeitstempel und Bildschirmabzügen eines VEE-Einblendfensters mit einer XY-Anzeige in einem MS Word-Dokument angezeigt werden können. In der Microsoft Dokumentation finden Sie ausführliche Beschreibungen darüber, wie MS Word mithilfe der ActiveX-Automatisierung über andere Anwendungen gesteuert werden kann.

Übung 6-3: Verwenden von MS Word für Agilent VEE-Berichte

Führen Sie zunächst die folgenden Schritte aus, um fünf Variablen als Typ `Object` zu deklarieren.

1. Klicken Sie `Device ⇒ ActiveX Automation References...` an und wählen Sie `Microsoft Word 9.0 Object Library` aus.
2. Klicken Sie `Data ⇒ Variable ⇒ Declare Variable` an.
   a. Ändern Sie das Typenfeld in `Object`. Klonen Sie dieses Objekt vier Mal.
   b. Benennen Sie die fünf Objektvariablen `App`, `Doc`, `Wnd`, `Sel` und `Bmp`.
   c. Wählen Sie `Specify Object Type` für alle fünf Objekte aus. Die Vorteile bei der Deklarierung der bestimmten `Class` innerhalb einer `Library` sind: VEE kann eine Typenprüfung auf Programmfehler ausführen, und Sie können Ereignisse vom Automatisierungs-Server erfassen.
   d. Klicken Sie die Schaltfläche `Edit...` an und wählen Sie in jedem Fall `Word for Library` aus. Wählen Sie die folgenden `Classes` aus:

      `App` verwendet `Application`
      `Sel` verwendet `Selection`
      `Wnd` verwendet `Window`
      `Doc` verwendet `Document`
      `Bmp` verwendet `Shape`

e. Wählen Sie `Enable Events` aus, sofern die jeweilige Klasse dies erlaubt. Stellen Sie die fünf Objekte als Symbole dar. Abbildung 6-14 zeigt die offene Ansicht dieser Variablen.

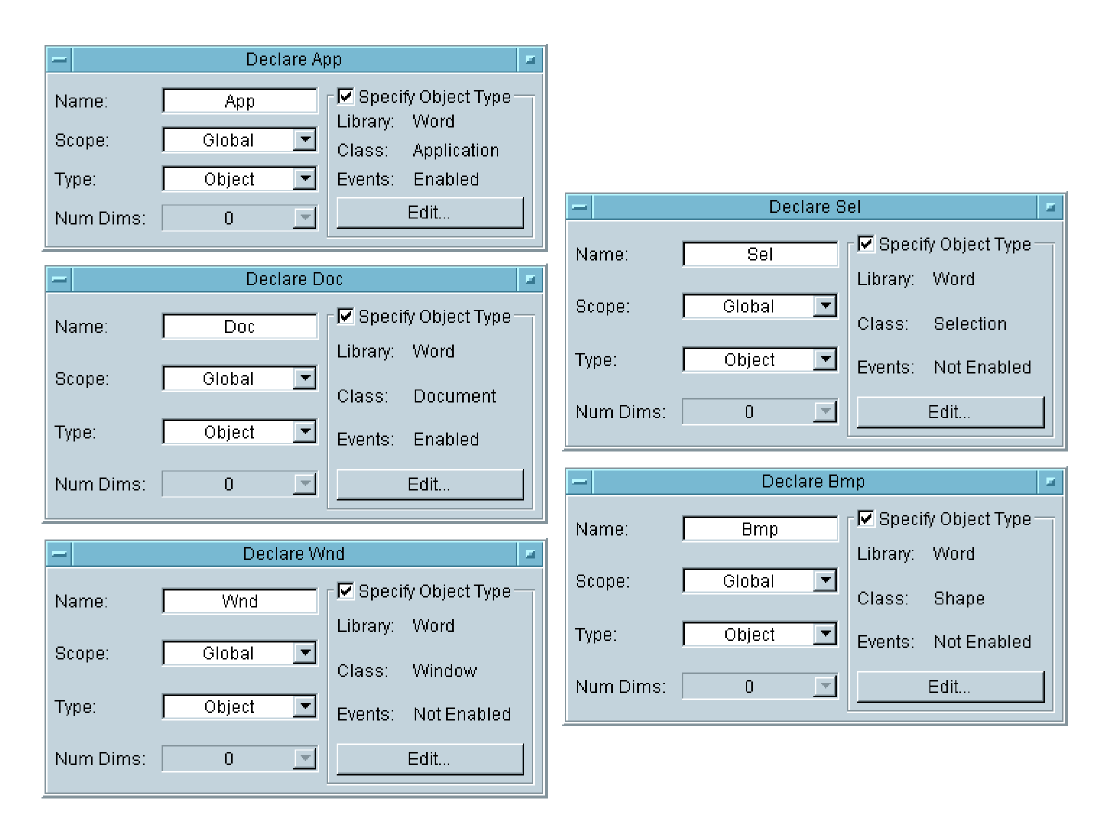


Abbildung 6-14. Objektvariablen

3. Erstellen Sie eine `UserFunction` mit dem Namen `Graph`, die eine virtuelle Quelle `Function Generator` zum Senden einer Sinuskurve an eine `Waveform (Time)`-Anzeige verwendet. Erstellen Sie eine Fensteransicht der Anzeige. Generieren Sie anschließend ein Objekt `Call Graph` im Hauptfenster. (Denken Sie daran, dass das `UserFunction`-Objektmenü eine einfache Möglichkeit zum Generieren eines Aufrufs bietet.)

Erstellen Sie jetzt eine Bitmap-Datei des Fensters mit der Waveform-Anzeige zur Verwendung in dem Bericht in MS Word.

4. Klicken Sie zum Erstellen eines Dateinamens für die Bitmap-Datei `Device ⇒ Formula` an. Benennen Sie das Objekt um in `Image Filename`. Geben Sie `installDir() + "\\panel.bmp"` im Eingabefeld `Formula` ein. (Verwenden Sie die Escape-Sequenz `\\` zum Angeben des ASCII-Zeichens `\`.) Löschen Sie den Eingangsanschluss `A`.

   Wenn Sie beispielsweise eine Installation in `c:\Program Files\Agilent\` durchgeführt haben, generieren Sie die folgende Textzeichenfolge am Ausgangs-Pin `Result`:
   `C:\Program Files\Agilent\VEE Pro 6.0\panel.bmp`.

5. Erstellen Sie ein weiteres `Formula`-Objekt und geben Sie `savePanelImage("Graph", FileName, 256) ein`. Benennen Sie den Eingangsanschluss um in `FileName`.

   Dadurch wird der Bildschirmabzug aus `UserFunction Graph` in der Datei `panel.bmp` in dem Installationsverzeichnis mit einer Farbtiefe von 256 pro Pixel gespeichert.

6. Erstellen Sie ein weiteres `Formula`-Objekt und geben Sie die folgende Anweisung ein:
   `Set App = CreateObject("Word.Application")`
   Dadurch wird MS Word gestartet und die Objektvariable `app` zugeordnet, sodass sie auf dieses Exemplar der Anwendung verweist. Löschen Sie den Eingangsanschluss `A`. Verbinden Sie `Call Graph`, `ImageFileName` und `savePanelImage` wie in Abbildung 6-15 gezeigt.

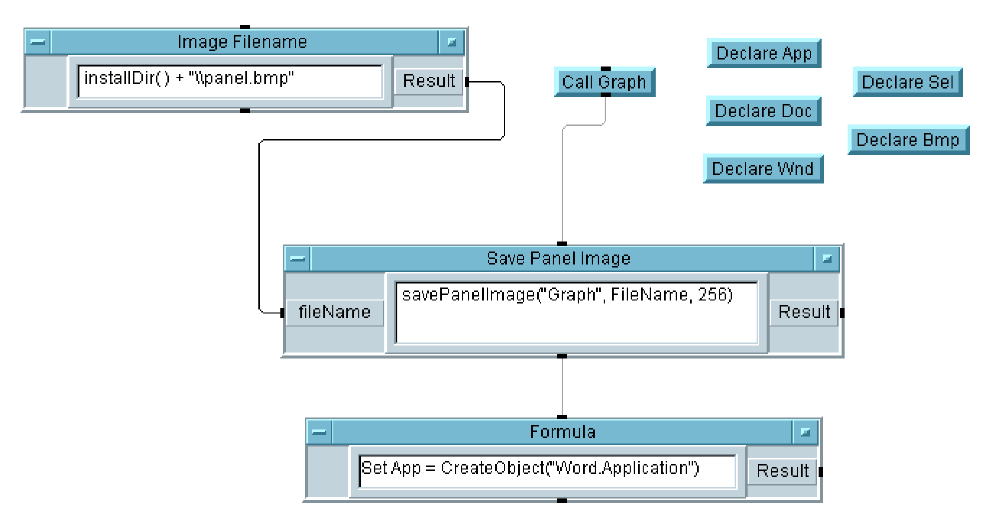


Abbildung 6-15. Anfang des Übungsprogramms 6-3

7. Klicken Sie `Device` ⇒ `Formula` an und geben Sie die in Abbildung 6-16 gezeigten Anweisungen ein; diese Anweisungen sind auch nachfolgend beschrieben. Benennen Sie den Eingangsanschluss `A` um in `FileName`. Verbinden Sie den Dateneingangs- und den Sequenzeingangs-Pin wie in Abbildung 6-16 gezeigt.

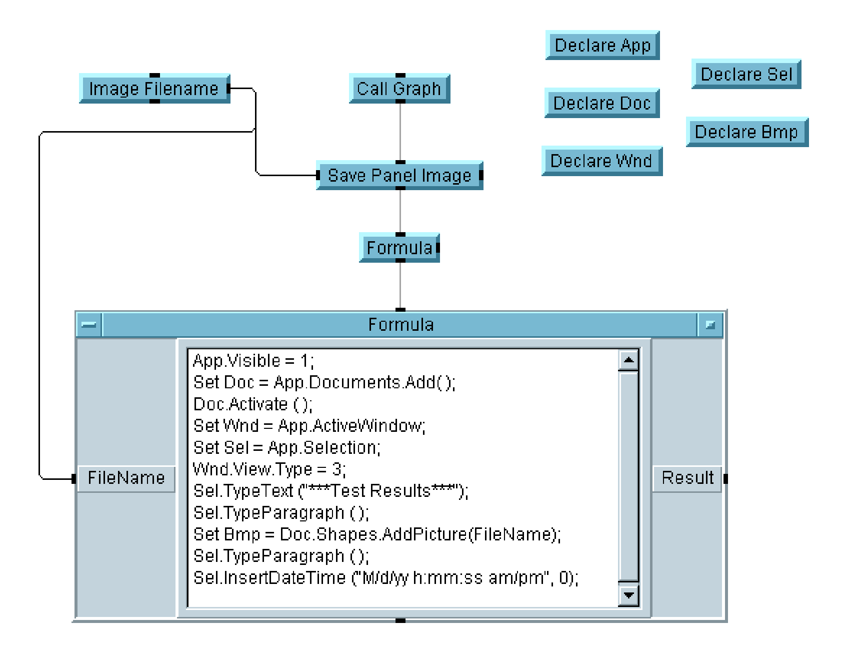


Abbildung 6-16. Hinzufügen der ActiveX-Anweisungen

Beachten Sie in Abbildung 6-16, dass Sie die Eigenschafts- und Methodenaufrufe mit der Punktnotation des `Objekts` verschachteln können. Schlagen Sie in der ActiveX-Dokumentation die richtigen Eigenschaften für die Zielanwendungen nach. Sie können die in diesem Kapitel beschriebenen Eigenschaften und Methoden verwenden, um mit dem Generieren von Test- und Messungsberichten zu beginnen. Die Einträge im `Formula`-Objekt lauten wie folgt:

App.Visible = 1; Macht MS Word auf dem Bildschirm sichtbar.

Set Doc = App.Documents. Add();	Fügt in MS Word ein Dokument hinzu und ordnet es der `Objekt`variablen `Doc` zu. Hinweis: Im Excel-Beispiel wurde Excel über `CreateObject(Excel.Sheet)` mit einer leeren Tabelle gestartet. In diesem Beispiel wird Word gestartet, und die Methode `Add()` fügt ein leeres Dokument hinzu. Beide Anwendungen können auf jede dieser beiden Arten erstellt werden.
Doc.Activate();	Aktiviert das Dokument.
Set Wnd = App.Active Window;	Verwendet das Dokument in dem aktiven Fenster und ordnet es der `Objekt`variablen `Wnd` zu.
Set Sel = App.Selection;	Setzt den Fokus (die Auswahl) auf das Dokument und ordnet es der `Objekt`variablen `Sel` zu. Dies ermöglicht die Eingabe von Text.
Wnd.View.Type = 3;	Gibt den Typ des Fensters zum Anzeigen des Dokuments an. Die 3 kennzeichnet eine normale Fenstergröße. Mit A 1 würde das Fenster als Symbol dargestellt. Hinweis: Hier wird "3" verwendet statt der Konstanten wdPageView, da die Konstante in der Typenbibliothek von Office 2000 nicht vorhanden ist.
Sel.TypeText(* Test Results ***), Sel.TypeParagraph();**	Fügt den Titel `*** Test Results ***` in das Dokument ein und gibt ein Zeilenvorschubzeichen aus.
Set Bmp = Doc.Shapes. AddPicture(FileName);	Legt das Bitmap `panel.bmp` in dem Dokument ab und ordnet diesen Aufruf in der `Shapes Class` der `Objekt`variablen `Bmp` zu.
Sel.TypeParagraph(); Sel.InsertDateTime (M/d/yy h:mm:ss am/pm, 0);	Legt einen Zeitstempel in dem Dokument ab.

8. Fügen Sie drei weitere `Formula`-Objekte und ein Objekt `If/Then/Else` hinzu. Konfigurieren und verbinden Sie diese Objekte wie in Abbildung 6-17 gezeigt.

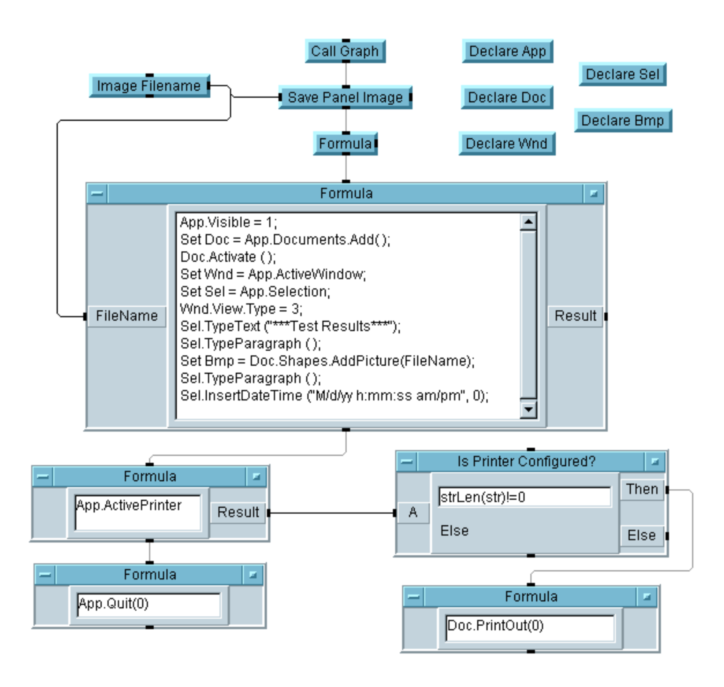


Abbildung 6-17. Das fertige Programm für den Bericht in MS Word

Die Einträge in den zusätzlichen Objekten lauten wie folgt:

App.ActivePrinter	Fordert den Standarddrucker an über eine Zeichenfolge, in der der entsprechende Port angegeben ist.
strLen(str) != 0	Stellt sicher, dass `ActivePrinter` einen konfigurierten Drucker gefunden hat ("if" - wenn die Zeichenfolge am Eingang nicht Null ist, "then" ...), gibt eine 1 `(=TRUE)` am Pin `Then` aus; dadurch erfolgt ein Ping auf das `Formula`-Objekt mit dem `PrintOut`-Aufruf.
DocPrintOut(0)	Druckt das Dokument.
App.Quit(0)	Schließt die Anwendung MS Word.

9. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 6-18. (Wenn die Farben in dem Bildschirmabzug merkwürdig aussehen, verkleinern Sie alle geöffneten Anwendungen auf Symbolgröße, damit dem PC die gesamte Farbpalette zur Verfügung steht.)

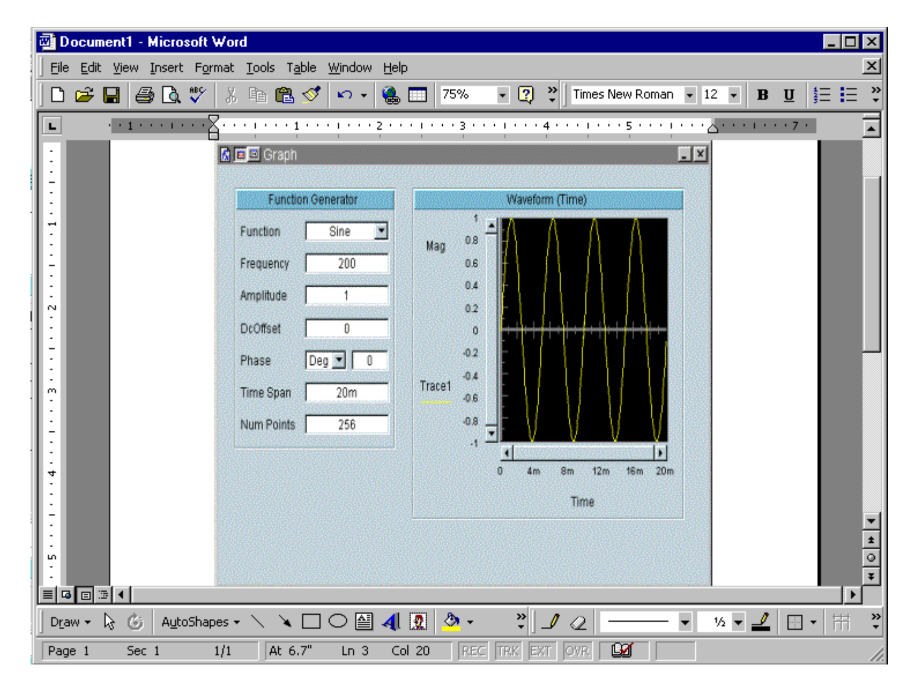


Abbildung 6-18. Das mit Übung 6-3 erstellte MS Word-Dokument

Weitere Informationen zum Steuern von MS Excel und MS Word über die ActiveX-Automatisierung finden Sie in der Microsoft Dokumentation. Denken Sie daran, dass Sie auch andere Server-Anwendungen, die die ActiveX-Automatisierung unterstützen, auf diese Weise steuern können. Die ActiveX-Automatisierung wird gelegentlich auch als Automation oder OLE-Automation bezeichnet.

Weitere Informationen zu ActiveX-Steuerelementen finden Sie in Kapitel 10, “Verwenden von Bedienerschnittstellen”. Weitere Informationen zur Verwendung von ActiveX über ein MS Visual Basic-Programm zum Steuern von VEE finden Sie in Kapitel 12, “Plattformspezifische Faktoren und Web-Überwachung”.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Beschreiben des Grundkonzepts hinter der ActiveX-Automatisierung in VEE.
- Senden von Daten von VEE an MS Excel.
- Verwenden einer allgemeinen Vorlage zum Senden von Arrays aus Testdaten an eine MS Excel-Tabelle. (Sie sollten wissen, wie Sie einen Array in einem einzigen Arbeitsschritt an eine Tabelle senden können.)
- Nutzen der erweiterten Funktionen der MS Excel-Bibliothek, z. B. zum Ermitteln des von einem Programm belegten Speichers.
- Senden von Text, Zeitstempel und Anzeige-Bitmap an MS Word von VEE aus.

7

Integrieren von Programmen in anderen Sprachen

Integrieren von Programmen in anderen Sprachen

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Das Objekt `Execute Program`
- Verwenden der Betriebssystemsbefehle von VEE aus
- Portierbar machen von VEE-Programmen über verschiedene Plattformen hinweg

Erforderliche Zeit für dieses Kapitel: 1 Stunde

Überblick

In diesem Kapitel lernen Sie die einfachste Möglichkeiten zur Integration von kompilierten Programmen und Betriebssystembefehlen mit VEE kennen. Einer der entscheidenden Vorteile von VEE ist die Möglichkeit zur Integration mit anderen Anwendungen und Programmen. Darüber hinaus können Sie über ActiveX Komponenten aus anderen Programmen verwenden. (Weitere Informationen hierzu finden Sie in Kapitel 6, “Einfaches Erstellen von Berichten mit ActiveX”.)

In VEE gibt das Objekt `Execute Program` Programme und Parameter an und verwendet Betriebssystembefehle. Es gibt ein Objekt `Execute Program` für den PC und ein anderes für HP-UX. Dieses Kapitel enthält eine Übung mit dem Objekt `Execute Program` für PCs und eine andere mit dem Objekt `Execute Program` für HP-UX.

Die Arbeitsweise des Objekts "Execute Program"

Zusätzlich zur ActiveX-Automatisierung gibt es drei weitere Möglichkeiten, Programme in anderen Sprachen von VEE aus auszuführen:

1. Verwenden des Objekts `Execute Program` zum Ausführung eines anderen Programms, einer anderen Anwendung oder eines Betriebssystembefehls außerhalb von VEE. Diese Methode ist besonders vielseitig und bedienerfreundlich.

2. Verbinden ("Link") kompilierter Funktionen in anderen Sprachen mit VEE, entweder über gemeinsame Bibliotheken in UNIX-Betriebssystemen oder Dynamic Link Libraries auf dem PC. Diese Methode ist zwar etwas komplizierter, bietet jedoch deutliche Vorteile bei der Geschwindigkeit. Weitere Informationen zu gemeinsamen Bibliotheken finden Sie im Abschnitt “Überblick über kompilierte Funktionen” auf Seite 437. Weitere Informationen zu Dynamic Link Libraries finden Sie im Abschnitt “Verwenden der Dynamic Link Libraries” auf Seite 440.

3. Verwenden einer speziell für Rocky Mountain Basic-Programme entwickelten Methode. Weitere Informationen hierzu finden Sie im Abschnitt “Kommunikation mit Rocky Mountain Basic-Programmen” auf Seite 466.

Das Objekt `Execute Program` befindet sich im Menü `I/O`. Es gibt ein Objekt für den PC und ein anderes für HP-UX (siehe Abbildung 7-1 und Abbildung 7-2). Beachten Sie, dass die PC-Version des Objekts `Execute Program` keine Transaktions-E/A zur Kommunikation mit Programmen verwendet; fügen Sie daher keine Dateneingangs- und -ausgangs-Pins zur Übergabe von Daten an das kompilierte Programm hinzu.

Die HP-UX-Version verwendet die Transaktions-E/A; das Beispiel enthält daher Ein- und Ausgangs-Pins und ein Programm zur Illustration der Arbeitsweise des Objekts.

Verwenden des Objekts "Execute Program" (PC)

Abbildung 7-1 zeigt das Objekt `Execute Program` auf dem PC.

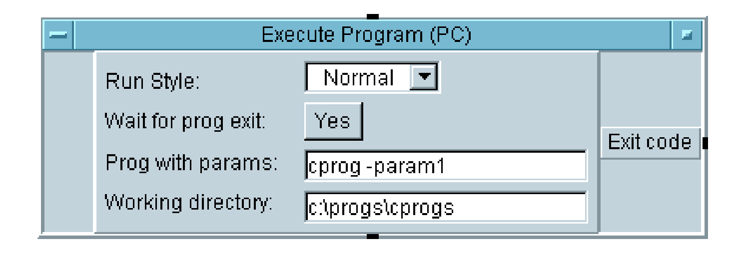


Abbildung 7-1. Das Objekt "Execute Program" (PC)

Verwenden Sie das Objekt `Execute Program` zur Ausführung folgender Elemente von VEE aus:

- Kompilierte Programme, die in anderen Sprachen geschrieben wurden
- `*.BAT`- oder `*.COM`-Dateien
- MS-DOS-Systembefehle, z. B. `dir`
- Beliebige Dokumente oder URLs mit einer bekannten Erweiterung. Die Aktion “open” wird in den Dateien aufgerufen. Wenn keine Aktion "open" vorhanden ist, wird die Standardaktion mit der Datei aufgerufen. Ein Beispiel für eine URL ist `http://www.agilent.com/find/vee`.

Die Felder im Objekt `Execute Program (PC)` sind:

Run Style Legt die Fenstergröße fest. `Normal` kennzeichnet ein Standardfenster, `Minimized` kennzeichnet ein Symbol, und `Maximized` kennzeichnet ein Fenster in Vollbildgröße. `Working directory` gibt das Arbeitsverzeichnis an, das die Dateien zu dem Programm enthält.

Wait for prog exit

Gibt an, wann der Sequenz-Pin auslösen soll.

- Wenn diese Angabe auf Yes gesetzt ist, wird der Sequenz-Pin nicht ausgelöst, bis die Ausführung des Programms abgeschlossen ist.
- Ist die Angabe auf No gesetzt, löst der Sequenz-Pin aus, bevor die Ausführung des angegebenen Programms abgeschlossen ist. Wenn beim Starten von Dokumenten oder URLs das Dokument oder die Web-Site in eine bereits aktive Anwendung geladen wird, wartet VEE nicht, bis die Anwendung beendet wird.

Prog with params

(Programm mit Parametern) Dieses Feld enthält die gleichen Wörter, die Sie in auch bei einer DOS-Eingabeaufforderung eingeben. Geben Sie zur Ausführung eines Programms in C beispielsweise den Namen der ausführbaren Datei
`- myprog.exe` ein. (Sie können die Erweiterung `.exe` weglassen.)

- Wenn das Programm Parameter hat, werden diese nach dem Namen der ausführbaren Datei und einem Bindestrich angegeben. Beispiel:
  `myprog -param1 -param2`.
- Zur Ausführung eines DOS-Systembefehls starten Sie zunächst den DOS-Befehlsinterpreter mit der Option `/c`. Für Windows 95 und Windows 98 geben Sie beispielsweise den Befehl `command.com /c` *<Systembefehl> ein.*
- Für Windows NT 4.0 und Windows 2000 geben Sie den Befehl `cmd /c` *<Systembefehl> ein.*
- Diese Option weist den Befehlsinterpreter an, die Zeichenfolge nach der Angabe `/c` als Systembefehl einzulesen.

Verwenden des Objekts "Execute Program" (HP-UX)

Abbildung 7-2 zeigt das Objekt `Execute Program` für HP-UX.

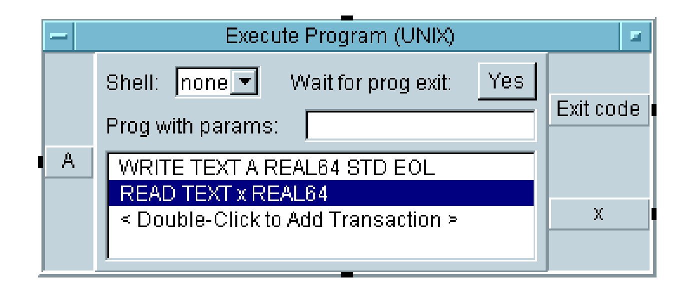


Abbildung 7-2. Das Objekt "Execute Program" (UNIX)

HP-UX ist für die gleichzeitige Ausführung mehrerer Programme ("Prozesse") konzipiert. Wenn VEE ein anderes Programm aufruft, wird VEE als "Elternprozess" (*parent*) bezeichnet und das aufgerufene Programm als "Kindprozess" (*child*). Das Objekt `Execute Program` initiiert ("spawn") einen Kindprozess entweder direkt oder über eine Befehls-Shell. Die Felder im Objekt `Execute Program (UNIX)` sind:

Shell

Das Feld `Shell` öffnet ein Menü mit den folgenden Optionen: `none`, `sh`, `csh` und `ksh`.

Das erste Kennzeichen im Feld `Prog with params` wird als der Name eines ausführbaren Programms interpretiert; die folgenden Kennzeichen werden als Parameter interpretiert.

Falls Sie im Feld `Prog with params` Shell-spezifische Funktionen verwenden wie beispielsweise die Standardeingabe oder die Ausgabeumleitung (`<<` und `>>`), Platzhalterzeichen (`*`, `?`, `[a-z]`) oder Pipes (|), müssen Sie eine Shell angeben; andernfalls wählen Sie `none` aus, da dies die Ausführung beschleunigt.

Wait for prog exit Das Feld `Wait for prog exit` schaltet um zwischen `Yes` und `No`. Unabhängig von der Einstellung initiiert VEE einen Kindprozess, falls nicht bereits ein solcher Prozess aktiv ist. Alle im Objekt `Execute Program` angegebenen Transaktionen werden abgeschlossen.

- Wenn diese Angabe auf `Yes` gesetzt ist, muss der Kindprozess abgeschlossen werden, bevor die Datenausgangs-Pins ausgelöst werden.
- Ist die Angabe auf `No` eingestellt, löst der Kindprozess die Datenausgangs-Pin aus und bleibt aktiv. Die Leistung Ihres Programms ist ohne diese Einstellung besser.

Prog with params Der Name einer ausführbaren Programmdatei und der Befehlszeilenparameter

-ODER-

ein Befehl, der zum Interpretieren an eine Shell gesendet wird.

Sie können dem Objekt `Execute Program` Ein- oder Ausgangsanschlüsse hinzufügen. Daten werden aus einem VEE-Programm an einem Eingangsanschluss empfangen; anschließend führen Sie eine `WRITE TEXT`-Transaktion aus, um die Daten an den Kindprozess zu senden. Eine `READ TEXT`-Transaktion liest Daten von dem Kindprozess ein und platziert sie an einem Datenausgangs-Pin zur Verwendung durch das VEE-Programm.

Sie können auch den Namen eines Programms oder Shell-Befehls an das Objekt `Execute Program` senden, indem Sie den Dateneingangsanschluss `Command` hinzufügen (verfügbar beim Hinzufügen von Dateneingängen).

Verwenden eines Systembefehls

Zum Aufrufen eines kompilierten Programms in einer anderen Sprache können Sie den Namen der ausführbaren Datei und alle erforderlichen Parameter am Objekt `Execute Program` eingeben.

Zur Ausführung eines MS-DOS-Systembefehls müssen Sie allerdings zunächst den DOS-Befehlsinterpreter starten. In dieser Übung starten Sie den DOS-Befehlsinterpreter und führen einen MS-DOS-Systembefehl aus.

Übung 7-1: Verwenden eines Systembefehls (PC)

1. Wählen Sie `I/O ⇒ Execute Program (PC)` aus. Klicken Sie das Feld `Prog with params` an, um einen Cursor anzuzeigen, und geben Sie anschließend ein:

   ```
   command.com /c dir >> c:\bob
   ```

Hinweis Ersetzen Sie `command.com` durch `cmd` für Windows NT 4.0 oder Windows 2000. Wenn der Laufwerksbuchstabe nicht `c:` ist, ersetzen Sie den Laufwerksbuchstaben in diesen Anleitungen entsprechend. Unter NT müssen Sie eventuell ein anderes Verzeichnis eingeben, für das Sie die Schreibberechtigung haben.

   (Eventuell müssen Sie den vollständigen Pfad der ausführbaren Datei `command.com` angeben.) Der Befehl startet den DOS-Befehlsinterpreter, der wiederum den Systembefehl zum Anzeigen des aktuellen Verzeichnisses startet und die Ausgabe (>) in die Datei `bob` statt an den Computerbildschirm umleitet.

   Behalten Sie `Yes` für die Auswahl `Wait for prog exit` bei. Behalten Sie `Normal` für `Run Style` bei und geben Sie `c:\` für das Arbeitsverzeichnis ein.

2. Wählen Sie `I/O ⇒ unter ⇒ File` aus und platzieren Sie dieses Objekt unter `Execute Program`. Verbinden Sie den Sequenzausgangs-Pin von `Execute Program` mit dem Sequenzeingangs-Pin des Objekts `From File`.

Klicken Sie das Eingabefeld `From File:` mit der Bezeichnung `myFile` an, um ein Listenfeld anzuzeigen, geben Sie `c:\bob` ein und klicken Sie `OK` an. (Das Programm erstellt die Datei `bob`.)

3. Klicken Sie doppelt auf die Transaktionsleiste, um das Feld `I/O Transaction` aufzurufen.

   a. Ändern Sie `REAL64 FORMAT` in `STRING FORMAT`.

   b. Ändern Sie `SCALAR` in `ARRAY 1D`.

   c. Klicken Sie das Feld `SIZE: (10)` an, um es auf `TO END: (*)` umzuschalten, und klicken Sie `OK` an. Die Transaktionsleisten sollte jetzt folgenden Inhalt haben: `READ TEXT x STR ARRAY:*`. Diese Transaktion liest den Inhalt der Datei `bob`.

4. Wählen Sie `Display` ⇒ `Logging AlphaNumeric` und verbinden Sie ihren Dateneingangs-Pin mit dem Datenausgang von `From File`.

5. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 7-3.

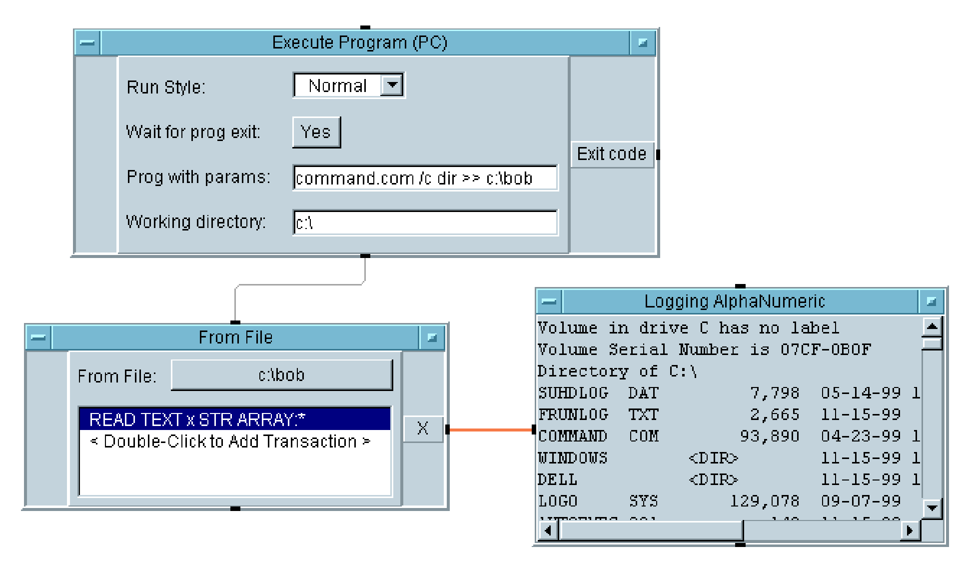


Abbildung 7-3. Auflisten der Dateien in einem Verzeichnis (PC)

Übung 7-2: Auflisten der Dateien in einem Verzeichnis (UNIX)

In dieser Übung verwenden Sie den Betriebssystembefehl `ls`, mit dem die Dateinamen in einem Verzeichnis aufgelistet werden. Da es sich hierbei nicht um einen shell-spezifischen Befehl handelt, können Sie `none` für die Angabe der Shell beibehalten. Anschließend programmieren Sie eine Variation dieser Übung mit einer shell-spezifischen Funktion, der Pipe (`|`).

Verwenden Sie zum Auflisten der Dateien in einem HP-UX-Programm den Befehl `ls` wie folgt.

1. Wählen Sie `I/O ⇒ Execute Program (UNIX)` aus und platzieren Sie dieses Objekt unten links im Arbeitsbereich.

2. Vergewissern Sie sich, dass das Feld `Shell` auf `none` eingestellt ist und das Feld `Wait for prog exit` auf `Yes`. Dadurch wird sichergestellt, dass der Befehl `ls` abgeschlossen ist, bevor VEE das Programm fortsetzt.

3. Klicken Sie das Feld `Prog with params` an und geben Sie ein: `ls /tmp`. (Sie können ein beliebiges Verzeichnis angeben.)

4. Fügen Sie einen Datenausgangsanschluss hinzu. Der Standardname lautet `X`. Es gibt keinen Exit-Code für das Programm; ignorieren Sie daher den Anschluss mit der Beschriftung `Exit code`.

5. Klicken Sie doppelt auf die Transaktionsleiste, um das Feld `I/O Transaction` aufzurufen. Ändern Sie die Standardvariable in ein `X`, da die Daten aus dem Programm an diesen Ausgangsanschluss eingelesen werden sollen.

   a. Ändern Sie `WRITE` in `READ`, um eine `READ TEXT`-Transaktion anzugeben.

   b. Ändern Sie `REAL64 FORMAT` in `STRING FORMAT`.

   c. Ändern Sie die Form der Daten von `SCALAR` in `ARRAY 1D`, um einen eindimensionalen Array zu kennzeichnen.

d. Schalten Sie SIZE: um auf TO END: (*), da Sie nicht wissen, wie viele Dateien sich in dem Verzeichnis befinden. Klicken Sie OK an. Die Transaktionsleiste sollte jetzt folgenden Inhalt haben READ TEXT X STR ARRAY:*.

6. Wählen Sie Display ⇒ Logging AlphaNumeric aus und verbinden Sie seinen Dateneingangs-Pin mit dem X-Anschluss am Objekt Execute Program.

7. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 7-4.

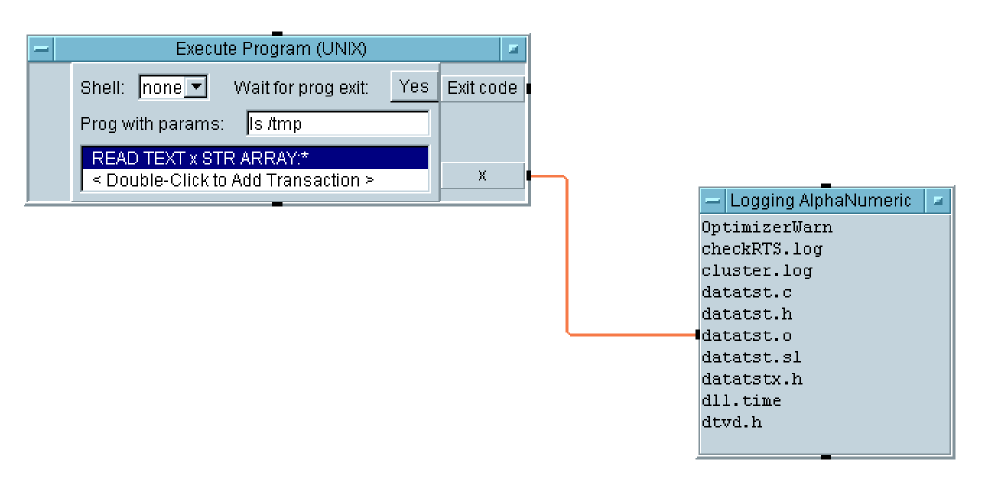


Abbildung 7-4. Auflisten der Dateien in einem Verzeichnis (UNIX)

Auflisten der Dateien in einem Verzeichnis mit einer Shell

Diese Variation der vorigen Übung verwendet eine shell-spezifische Funktion, die Pipe (|), zum Senden der Ausgabe eines Betriebssystembefehls an einen anderen. Der zweite Befehl ist wc; dies steht für word count. Der Befehl wc zählt Zeilen, Wörter und Zeichen in benannten Dateien. Der Befehl "wc -l Dateiname" zählt die Anzahl der Zeilen in der angegebenen Datei. Dieses Beispiel beschreibt, wie Sie die Anzahl von Dateien in einem Verzeichnis zählen und anschließend diese Anzahl und die Dateien anzeigen können.

1. Wählen Sie `I/O ⇒ Execute Program (UNIX)` aus.

    a. Setzen Sie das Feld `Shell` auf `sh`, um die Shell-Funktionen “`|`” und “`;`” zu verwenden.

    b. Geben Sie den Befehl `ls /tmp|wc -l;ls /tmp` im `Shell`-Befehlsfeld ein.

    c. Fügen Sie zwei Datenausgangsanschlüsse hinzu, einen mit der Bezeichnung `X` und den anderen mit der Bezeichnung `Lines`.

    d. Konfigurieren Sie die erste Transaktion als `READ TEXT Lines INT32`. "Lines" ersetzt die Standardvariable `a`.

    e. Konfigurieren Sie eine zweite Transaktion als `READ TEXT X STR ARRAY:Lines`. Geben Sie `Lines` im Feld `SIZE` ein, um die Länge des Array zu definieren.

2. Wählen Sie `Display ⇒ AlphaNumeric` aus und klonen Sie dieses Objekt. Verbinden Sie ein `AlphaNumeric`-Objekt mit dem Ausgang von `Lines` und ein anderes mit dem Ausgangs-Pin `X`.

3. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 7-5.

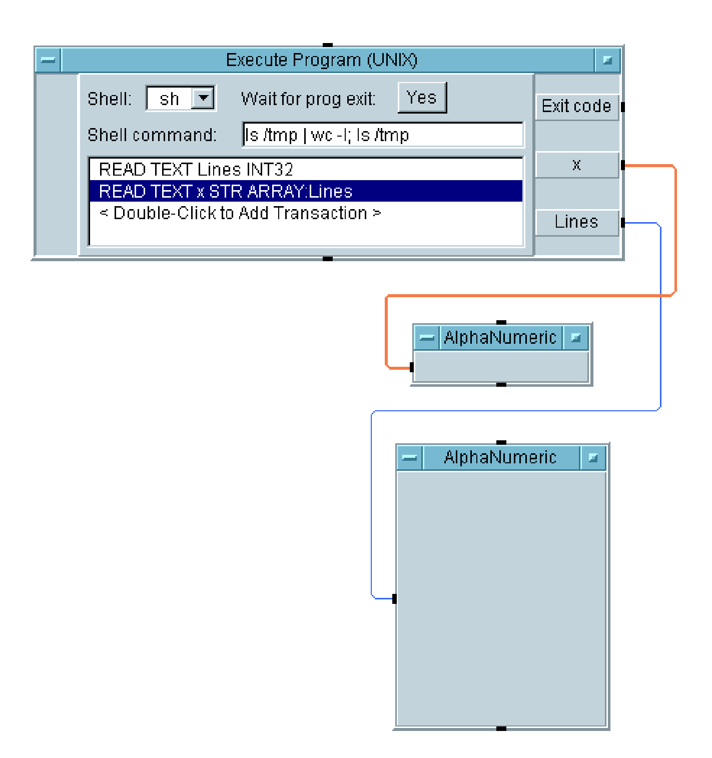


Abbildung 7-5. Verwenden eines Shell-Befehls mit einer Pipe

Schreiben von einfach portierbaren Programmen

Wenn Sie Programme in anderen Sprachen integrieren wollen, schreiben Sie die VEE-Programme so, dass sie leicht über verschiedene Plattformen hinweg portiert werden können. VEE umfasst Systeminformationsobjekte in `Function & Object Browser` ⇒ `System Information` (siehe Abbildung 7-6). Diese Objekte können auch als Funktionen verwendet werden.

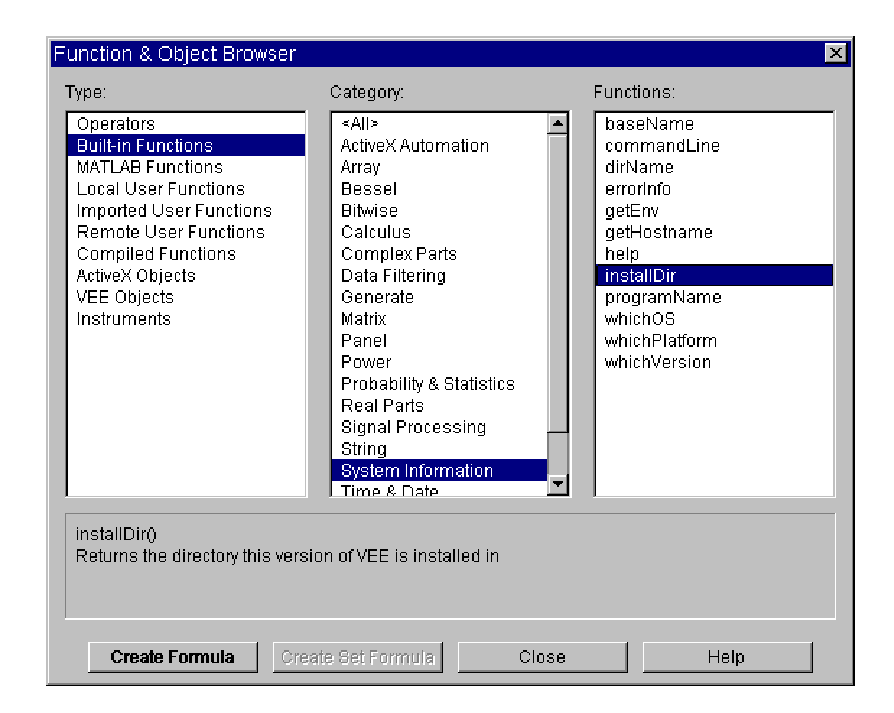


Abbildung 7-6. Systeminformationsfunktionen

Die zur Verbesserung der Portierbarkeit von Programmen verwendeten allgemeinen Systeminformationsfunktionen im `Function &Object Browser` sind:

installDir Gibt den Namen des VEE-Installationsverzeichnisses an.

whichOS	Gibt das Betriebssystem an und sendet eine der folgenden Zeichenfolgen: `Windows_95`, `Windows_98`, `Windows_2000`, `Windows_NT`, `HP-UX`. Das Programm kann beim Integrieren von Programmen aus anderen Sprachen entsprechend diesen Ergebnissen verzweigen. Sehen Sie sich beispielsweise mit `manual49.vee` im Verzeichnis `examples\manual` ein Programm an, das `whichOS()` verwendet, um sicherzustellen, dass es den richtigen Bibliothekstyp importiert. Unter HP-UX würde es eine gemeinsame Bibliothek importieren. Auf einem PC dagegen würde es eine Dynamic Link Library importieren.
whichplatform	Legt das Hardwaresystem fest, auf dem VEE ausgeführt wird, und gibt anschließend eine Zeichenfolge zurück, die diese Plattform kennzeichnet.
whichVersion	Gibt die VEE-Version an. Dies ist für die Programmpflege und die Fehlerbehebung hilfreich.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Erläutern des Zwecks des Objekts `Execute Program`.
- Überblick über die Konfigurationseinstellungen am Objekt `Execute Program`.
- Erläutern des allgemeinen Prozesses, mit dem das Objekt `Execute Program` Daten von einem Programm an eine PC-Plattform sendet und umgekehrt.
- Erläutern, wie sich die HP-UX-Plattform von der PC-Plattform unterscheidet.
- Ausführen von Betriebssystembefehlen von VEE aus.
- Erstellen eines Programms, das die Objekte `whichOS()`, `whichPlatform()` oder `whichVersion()` verwendet, sodass es unter verschiedenen Betriebssystemen verwendet werden kann.

8

Verwenden von Agilent VEE-Funktionen

Verwenden von Agilent VEE-Funktionen

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Definieren einer Benutzerfunktion
- Erstellen, Aufrufen und Bearbeiten von Funktionen
- Erstellen, Zusammenführen, Importieren und Löschen von Funktionsbibliotheken
- Finden von Funktionen in großen Programmen
- Zusammenführen vorhandener VEE-Programme mit Tests

Erforderliche Zeit für dieses Kapitel: 1 Stunde

Überblick

In diesem Kapitel erfahren Sie mehr über VEE `UserFunctions`, kompilierte Funktionen und Fernfunktionen. Funktionen sind wiederverwendbare, modulare Codeelemente, mit denen Sie die für die Testentwicklung erforderliche Zeit erheblich verkürzen können. Durch die Wiederverwendung von Funktionen, die bereits in anderen Programmen erstellt wurden, können Sie von bereits erledigten Arbeiten profitieren, die Codegröße Ihrer Programme reduzieren und die Pflege Ihrer Testprogramme erleichtern. Sie können Funktionen auch als Gruppen, den so genannten Bibliotheken, verwenden. Bibliotheken können Sie erstellen und in neue Programme einbinden. Sie können Funktionen über verschiedene Programme und Hersteller hinweg gemeinsam nutzen.

Verwenden von Funktionen

Wie viele andere Programmiersprachen verwendet VEE Funktionen zum Erstellen von Unterprogrammen, die bestimmte Aufgaben ausführen. Die Übungen in diesem Kapitel beschreiben das Erstellen, Aufrufen und Ändern von benutzerdefinierten VEE-Funktionen. Außerdem lernen Sie, wie Sie Bibliotheken von Funktionen erstellen, die Sie in der Entwicklungsphase oder zur Ausführungszeit in Programme einbinden können.

Definieren einer Agilent VEE-Funktion

Es gibt drei Arten von benutzerdefinierten Funktionen in VEE. Nachfolgend finden Sie einen Überblick über die verschiedenen Funktionstypen:

1. `UserFunctions`

   - ❑ Zum *Erstellen* einer `UserFunction` wählen Sie `Device ⇒ UserFunction` aus, oder klicken Sie `Edit ⇒ Create UserFunction` an, während verschiedene Objekte ausgewählt sind.

   - ❑ Zum *Aufrufen* einer `UserFunction` von verschiedenen Stellen in einem Programm verwenden Sie das Objekt `Call myFunction` (Device ⇒ Call) oder einen Ausdruck in einem Objekt (beispielsweise von `Formula` aus). Sie können auch Aufrufobjekte m Hauptprogramm von der `UserFunction` aus generieren mit dem `UserFunction`-Objektmenü und der Auswahl von Optionen wie `Generate ⇒ Call`.

   - ❑ Zum *Ändern* einer `UserFunction` klicken Sie `Edit ⇒ Edit UserFunction...` an und wählen die entsprechende `UserFunction` in dem angezeigten Listenfeld aus.

   - ❑ Zum *Übertragen von* `UserFunctions` aus einem Programm in ein anderes kombinieren Sie die `UserFunctions` bei der Programmentwicklung oder importieren sie zur Ausführungszeit (`Device ⇒ Import Library`).

2. Kompilierte Funktionen

   - Zum *Erstellen* einer kompilierten Funktion arbeiten Sie außerhalb von VEE mit einer kompilierten Sprache. Anschließend legen Sie die Funktionen in einer Bibliothek wie beispielsweise einer DLL ab.

   - Zum *Verbinden* ("Link") einer kompilierten Funktion mit einem Programm verwenden Sie das Objekt `Import Library`, mit dem die Bibliothek zur Ausführungszeit mit VEE verbunden wird. (Eine ausführlichere Beschreibung finden Sie in Kapitel 11, “Optimieren von Agilent VEE-Programmen.”).

   - Zum *Aufrufen* einer kompilierten Funktion verwenden Sie das Objekt `Call myFunction` oder schreiben einen Ausdruck in einem VEE-Objekt.

3. Fernfunktionen

   - Fernfunktionen ähneln `UserFunctions`, werden jedoch auf einem fernen an Ihr Netzwerk angeschlossenen fernen Host-Computer ausgeführt.

Die Unterschiede zwischen UserObjects und UserFunctions

In den vorigen Kapiteln haben Sie bereits `UserObjects` erstellt und verwendet. Der Grund, dass VEE sowohl `UserObject` als auch `UserFunction` zur Verfügung stellt, liegt darin, dass die beiden Komponenten unterschiedliche Merkmale aufweisen und daher für verschiedene Zwecke eingesetzt werden können. Die Unterschiede zwischen einem `UserObject` und einer `UserFunction` sind:

Ein `UserObject` (in `Device ⇒ UserObject`) ist ein von Ihnen definiertes Objekt, das genau wie jedes andere Objekt in VEE verwendet werden kann. Sie programmieren ein `UserObject` wie ein Unterprogramm; es bleibt jedoch grafisch auf der Anzeige erhalten. Wenn Sie es an einer anderen Stelle im Programm verwenden wollen, müssen Sie es klonen und alle Kopien verwalten. Wenn Sie ein `UserObject` oft klonen, wird das Programm dadurch größer und erfordert eine längere Ladezeit. Wenn Sie einem

UserObject eine Funktion hinzufügen, müssen Sie diese Funktion auch allen anderen UserObjects hinzufügen, wenn diese Objekte weiterhin identisch sein sollen.

Von einer UserFunction (in Device ⇒ UserFunction) ist dagegen nur ein einziges Exemplar im Speicher vorhanden. Eine UserFunction wird nur dann im Arbeitsbereich in einem eigenen Fenster grafisch dargestellt, wenn Sie dies ausdrücklich angeben. Andernfalls wird sie gespeichert und über das Objekt Call oder ein anderes Ausdrucksfeld aufgerufen. Änderungen an einer UserFunction werden von allen Exemplaren in dem Programm, das diese UserFunction aufruft, übernommen. Sie können auch Bibliotheken von UserFunctions für eine zusätzliche Wiederverwendung des Codes erstellen.

Übung 8-1: UserFunction-Operationen

In dieser Übung wird beschrieben, wie Sie eine UserFunction mit dem Namen ArrayStats erstellen, die einen Array akzeptiert, zu diesem Array den Maximalwert, den Minimalwert, den Durchschnittswert und die Standardabweichung berechnet und die Ergebnisse an eigenen Ausgangs-Pins anlegt.

Erstellen einer UserFunction

1. Wählen Sie Device ⇒ Formula aus, löschen Sie ihren Standard-Eingangs-Pin und ändern Sie ihren Ausdruck in ramp(1024,1,1024).

   Dadurch wird ein Array aus 1024 Elementen mit Werten von 1 bis 1024 erstellt.

2. Wählen Sie Device ⇒ UserFunction aus. Benennen Sie das Objekt um in ArrayStats.

   a. Fügen Sie einen Dateneingangsanschluss für den Array hinzu.

   b. Fügen Sie vier Datenausgangsanschlüsse für die Ergebnisse hinzu.

   c. Benennen Sie die Ausgangsanschlüsse um: Max, Min, Mean und Sdev. Wählen Sie max, min, mean und sdev in der Kategorie

`Probability & Statistics` im Fenster `Function & Object Browser` aus.

d. Platzieren Sie sie in `ArrayStats` und verbinden Sie ihre Dateneingänge mit `A` und ihre Datenausgänge mit den entsprechenden Ausgangsanschlüssen. Verkleinern Sie das Fenster `ArrayStats`, um sowohl das Hauptfenster als auch das Fenster `ArrayStats` anzuzeigen. Siehe Abbildung 8-1.

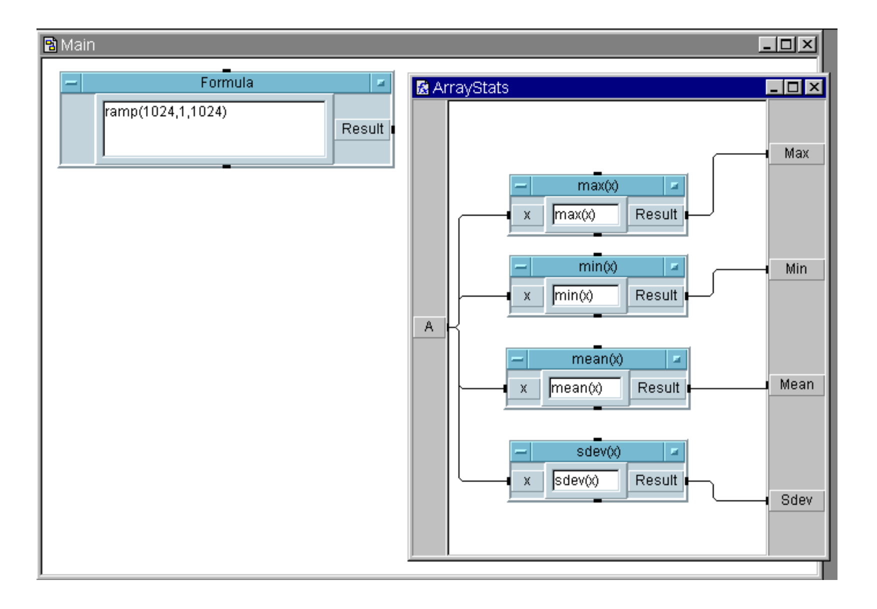


Abbildung 8-1. Hauptfenster und Fenster "ArrayStats"

3. Stellen Sie `ArrayStats` als Symbol dar. Das Symbol wird am unteren Rand des Arbeitsbereichs angezeigt.

4. Klicken Sie `Device` ⇒ `Call` an, öffnen Sie das Objektmenü und klicken Sie `Select Function` an, wie in Abbildung 8-2 gezeigt. Klicken Sie anschließend `OK` an. Beachten Sie, dass VEE das Objekt automatisch umbenennt und die richtigen Pins hinzufügt.

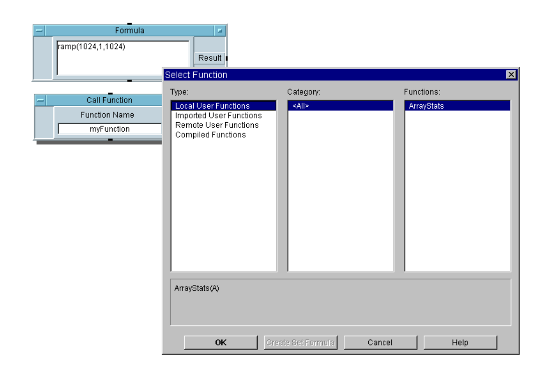


Abbildung 8-2. Konfigurieren der Pins für Call myFunction

5. Verbinden Sie den Ausgang von `Formula` mit dem Eingang von `Call ArrayStats`. Wählen Sie `Display` ⇒ `AlphaNumeric` aus, klonen Sie dieses Objekt drei Mal und verbinden Sie die Anzeigen mit den Ausgangs-Pins von `Call ArrayStats`. Benennen Sie die Anzeigen um.

6. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 8-3. Speichern Sie das Programm als `array_stats.vee`.

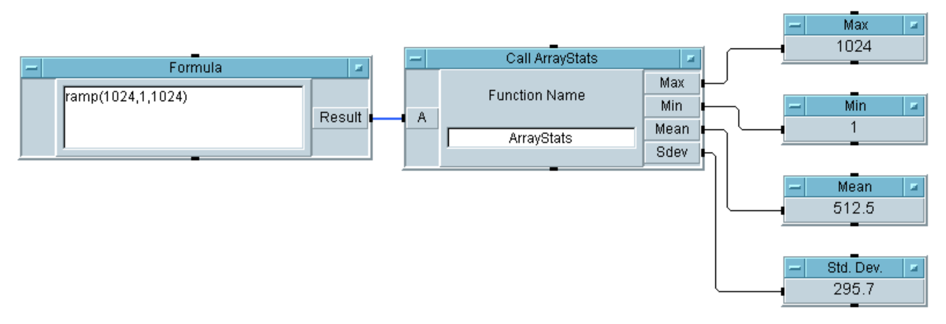


Abbildung 8-3. Aufruf der User Function ArrayStats

Zur Verwendung von `ArrayStats` an einer anderen Stelle im Programm klicken Sie `Device` ⇒ `Call` an, öffnen das Fenster `Select Function` über das Objektmenü und wählen `ArrayStats` aus. VEE benennt in diesem Fall das Objekt `Call ArrayStats` automatisch um und fügt die erforderlichen Ein- und Ausgangsanschlüsse hinzu.

Shortcut: Wählen Sie im `UserFunction`-Objektmenü `Generate ⇒ Call` aus, um das Objekt `Call ArrayStats` zu öffnen. (Vergewissern Sie sich, dass bei dieser Aktion die `UserFunction` nicht auf den gesamten Arbeitsbereich erweitert wurde.)

Ändern einer UserFunction

In dieser Übung ändern Sie `ArrayStats` so, dass es einen Datensatz mit vier Feldern liefert, die die Array-Statistik angeben.

1. Löschen Sie die vier `AlphaNumeric`-Anzeigen.

2. Wählen Sie `Edit ⇒ Edit UserFunction...` und anschließend `ArrayStats` im Listenfeld `Edit UserFunction` aus. Alle `UserFunctions` in dem Programm werden angezeigt.

3. Öffnen Sie das `ArrayStats`-Objektmenü, klicken Sie `size` an und vergrößern Sie das Bearbeitungsfenster. Wenn Sie die Größe der Objekte ändern müssen, klicken und ziehen Sie eine beliebige Ecke des Objekts.

4. Löschen Sie die vier Linien zu den Ausgangsanschlüssen. (Drücken Sie die Tastenkombination **Strg-Umschalttaste** und klicken Sie auf die Linie, die Sie löschen wollen.)

5. Wählen Sie `Data ⇒ Build Data ⇒ Record` aus und platzieren Sie das Objekt rechts vom Fenster `ArrayStats`.

   a. Fügen Sie zwei Dateneingangsanschlüsse hinzu.

   b. Beschriften Sie die vier Anschlüsse nach den statistischen Funktionen: `Max`, `Min`, `Mean` und `Sdev`.

   c. Verbinden Sie die vier Ausgänge des `Formula`-Objekts mit den Eingängen an `Build Record`.

   d. Benennen Sie den Ausgangsanschluss `X` an `Max` um, indem Sie doppelt auf `Max` klicken, den neuen Namen eingeben und `OK` anklicken.

e. Löschen Sie die weiteren Datenausgangsanschlüsse von `ArrayStats`.

f. Verbinden Sie den Ausgang von `Build Record` mit dem Ausgangsanschluss `X` des `User Function`-Bearbeitungsfensters. Das Programm sollte jetzt aussehen wie in Abbildung 8-4. Klicken Sie anschließend die Schaltfläche "Symbolgröße" in dem Fenster an.

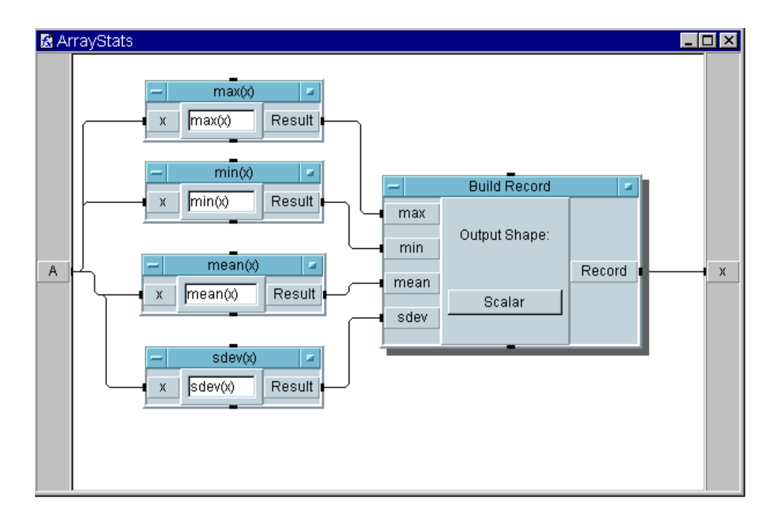


Abbildung 8-4. Ändern der UserFunction ArrayStats

6. Öffnen Sie das `Call ArrayStats`-Objektmenü und klicken Sie `Configure Pinout` an. Dadurch wird die Anzahl der Pins entsprechend den letzten Änderungen angepasst.

Hinweis

Zum Aktualisieren der Anzahl von Pins müssen Sie das Objekt öffnen und `Configure Pinout` anklicken, wann immer Sie die Anzahl von Eingängen und/oder Ausgängen in einer `UserFunction` ändern. Sie können die Ein- und Ausgangs-Pins des `Call`-Objekts auch manuell aktualisieren; die Verwendung von `Configure Pinout` ist jedoch erheblich einfacher. Sie können mit `Find` alle `Call`-Objekte und Ausdrücke finden, die eine `UserFunction` aufrufen. Weitere Informationen hierzu finden Sie im Abschnitt “Finden von Funktionen in großen Programmen” auf Seite 337.

Zeigen Sie jetzt einen Datensatz mit dem Objekt `Record Constant` an. Verwenden Sie den Steuereingang `Default Value` zum Akzeptieren eines Datensatzes von `ArrayStats`. VEE konfiguriert das Objekt `Record Constant` automatisch zum Speichern des ankommenden Datensatzes.

7. Wählen Sie `Data ⇒ Constant ⇒ Record` und platzieren Sie dieses Objekt rechts von dem Objekt `Call Function`.

   a. Öffnen Sie das `Record`-Objektmenü und klicken Sie `Add Terminal ⇒ Control Input...` an. Wählen Sie `Default Value` im angezeigten Listenfeld aus. Sie können das Menü `Properties` öffnen, um die Anschlüsse anzuzeigen (`Show Terminals`).

   b. Verbinden Sie jetzt den Datenausgang von `Call Function` mit dem Steuereingangs-Pin am `Record`-Objekt. Steuerlinien werden durch gestrichelte Linien gekennzeichnet, um sie von Datenlinien zu unterscheiden.

8. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 8-5.

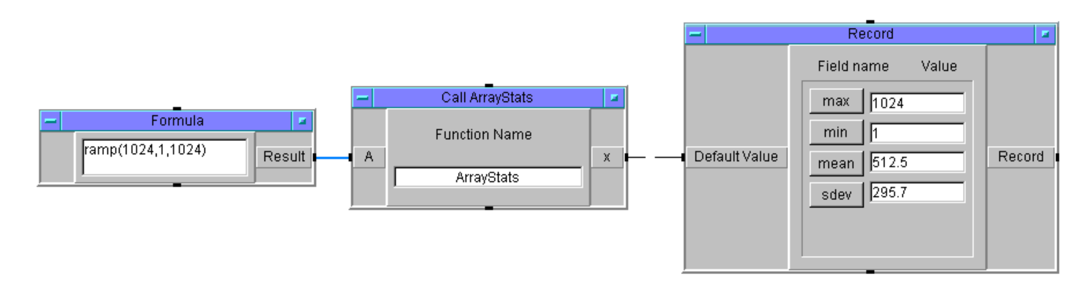


Abbildung 8-5. Ausgabe an einen Datensatz nach dem Ändern von ArrayStats

Aufrufen einer UserFunction von einem Ausdruck

In dieser Übung lernen Sie, wie Sie `ArrayStats` von einem Ausdruck im `Formula`-Objekt aus aufrufen.

1. Wählen Sie `Device ⇒ Formula` aus und ersetzen Sie die Standardformel durch `ArrayStats(A)`. Klicken Sie `Replace` im `Call ArrayStats`Objektmenü an.

   Die Statusleiste am unteren Rand der VEE-Anzeige fordert Sie auf, das Ersatzobjekt auszuwählen. Klicken Sie das `Formula`-Objekt an, das die `ArrayStats`-Funktion aufruft. VEE ersetzt das `Call ArrayStats`-Objekt automatisch durch das neue `Formula`-Objekt und behält die Datenlinien bei.

   Das `Formula`-Objekt verwendet den Eingang am Anschluss `A` und sendet ihn an die `UserFunction ArrayStats`. `ArrayStats` liefert den Datensatz der Statistik an seinen Anschluss `X`. Der erste Ausgangswert von der `UserFunction (X)` wird an das `Formula`-Objekt zurückgegeben und an seinen `Result`-Ausgang geliefert.

2. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 8-6.

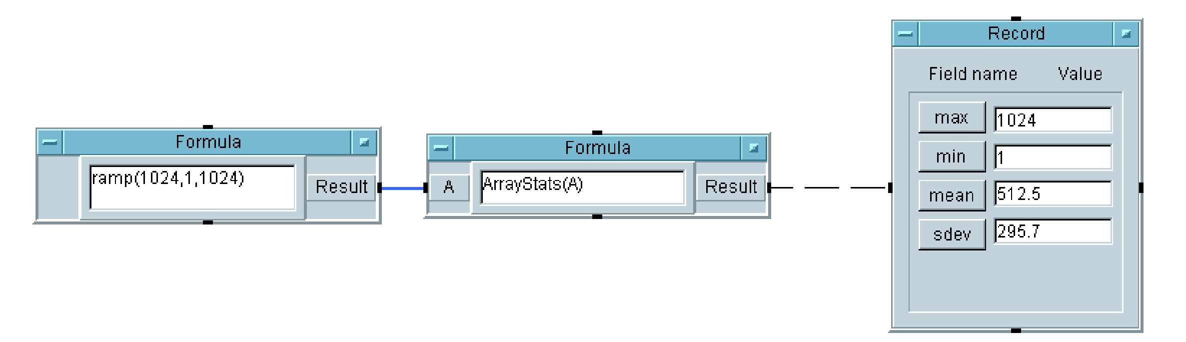


Abbildung 8-6. Aufrufen der User Function ArrayStats

Beachten Sie, dass die Funktionalität von `ArrayStats` im Objekt `Formula` exakt dieselbe ist wie zuvor im Objekt `Call ArrayStats`. Dieses Beispiel verwendet ein `Formula`-Objekt; Sie können jedoch `ArrayStats` von *jedem beliebigen* Eingabefeld aus aufrufen, das Ausdrücke akzeptiert, z. B. das Objekt `To File`.

Hinweis

Wenn Sie eine `UserFunction` von einem Ausdruck aus aufrufen, liefert die `UserFunction` nur einen einzigen Ausgang (den obersten Datenausgangs-Pin). Wenn Sie alle Ausgänge benötigen oder sie nicht in einem `Record` abgelegt werden können, verwenden Sie das Objekt `Call Function`.

Hinweis

Wenn Sie eine `UserFunction` von einem Ausdruck aus aufrufen, werden Eingangsanschlüsse als Funktionsparameter verwendet, um die Funktion weiterzugeben. Auch wenn keine Daten an die Funktion übergeben werden, müssen Sie leere Klammern nach dem Funktionsnamen angeben. Andernfalls geht VEE davon aus, dass Sie sich auf eine `globale` Variable oder einen Eingangsanschluss beziehen. Wenn die `UserFunction` mit dem Namen `MyFunction` beispielsweise keine Eingangsparameter hat, müssen Sie `MyFunction()` in einem Ausdruck angeben. Das `Call`-Objekt erfordert hingegen keine Klammern, da VEE weiß, dass Sie sich auf eine Funktion beziehen.

Generieren eines Aufrufs einer UserFunction

Zum Generieren und Platzieren eines Call-Objekts im Hauptprogramm von einer `UserFunction` aus verwenden Sie das `UserFunction`-Objektmenü `Generate`. Das Menü `Generate` enthält die meisten gängigen Objekte, die eine `UserFunction` aufrufen. Wenn Sie ein aufrufendes Objekt auswählen, kann es mit der richtigen Konfiguration mit Namen und Pins in dem aufrufenden Fenster platziert werden.

In dieser Übung lernen Sie, wie Sie das Objekt `ArrayStats` im Hauptprogramm von der `UserFunction ArrayStats` aus generieren.

1. Klicken Sie im gleichen Beispiel in Abbildung 8-6 doppelt auf das `Formula`-Objekt `ArrayStats`, um das Objekt zu löschen. (Sie können auf das Objektmenü auswählen und "Cut" auswählen.)

2. Wählen Sie in der `UserFunction ArrayStats` das Objektmenü und anschließend `Generate` ⇒ `Formula Call` aus. Abbildung 8-7 zeigt das Menü `Generate` in einem `UserFunction`-Objektmenü.

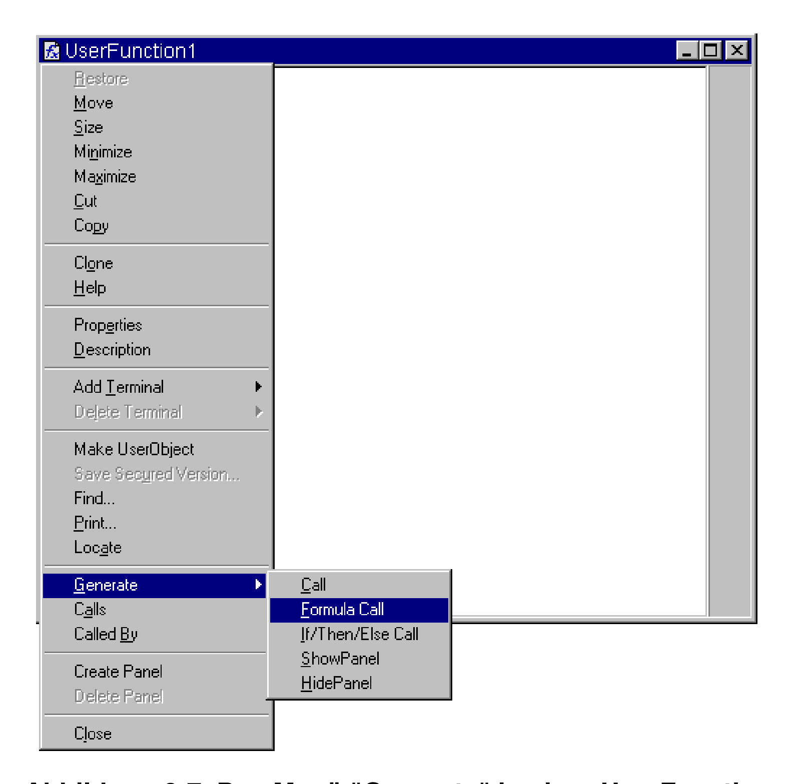


Abbildung 8-7. Das Menü "Generate" in einer UserFunction

3. Platzieren Sie das Objekt im Hauptprogramm. Beachten Sie, dass VEE das neue Objekt automatisch `ArrayStats(A)` benennt und den Ausdruck `ArrayStats(A)` zum Aufruf der `UserFunction ArrayStats` einbindet.

4. Verbinden Sie den Ausgang des `Formula`-Objekts mit `ArrayStats(A)` und den Ausgang von `ArrayStats(A)` mit `Record`.

5. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 8-8.

Öffnen Sie ein `UserFunction`-Objektmenü und wählen Sie das Menü `Generate` aus, um die anderen Objekte anzuzeigen, die in einem Programm zum Aufrufen einer `UserFunction` platziert werden können. Hierzu gehören die Objekte `Call`, `Formula Call` (in diesem Beispiel verwendet), `If/Then/Else Call`, `ShowPanel` und `HidePanel`.

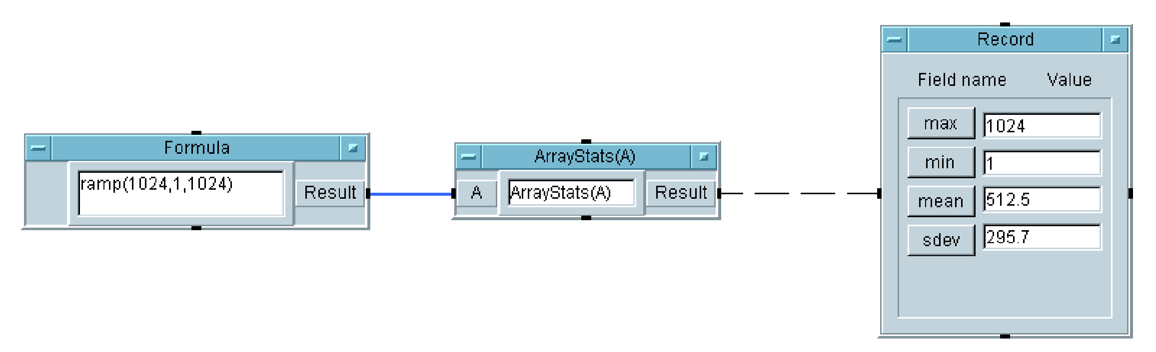


Abbildung 8-8. Generieren eines Aufrufs des Objekts ArrayStats(A) von einer UserFunction aus

UserFunctions und der Program Explorer

`UserFunctions` und `UserObjects` gestalten VEE-Programme modularer und machen sie leichter verständlich. Der Program Explorer ist ein hilfreiches Tool zum Navigieren durch komplexe Programme. Die Hierarchie des Programms in Abbildung 8-9 wird beispielsweise im Program Explorer angezeigt. Zum Anzeigen des Program Explorer klicken Sie `View` ⇒ `Program Explorer` oder klicken das Symbol **Program Explorer** in der Symbolleiste an.

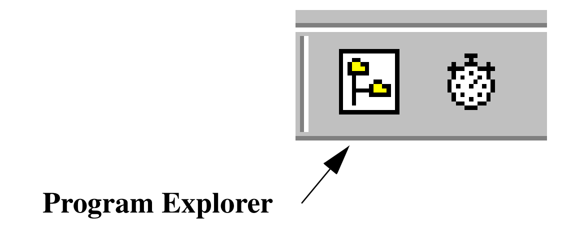


Abbildung 8-9. Symbol "Program Explorer" in der Symbolleiste

Abbildung 8-10 zeigt den verwendeten Program Explorer.

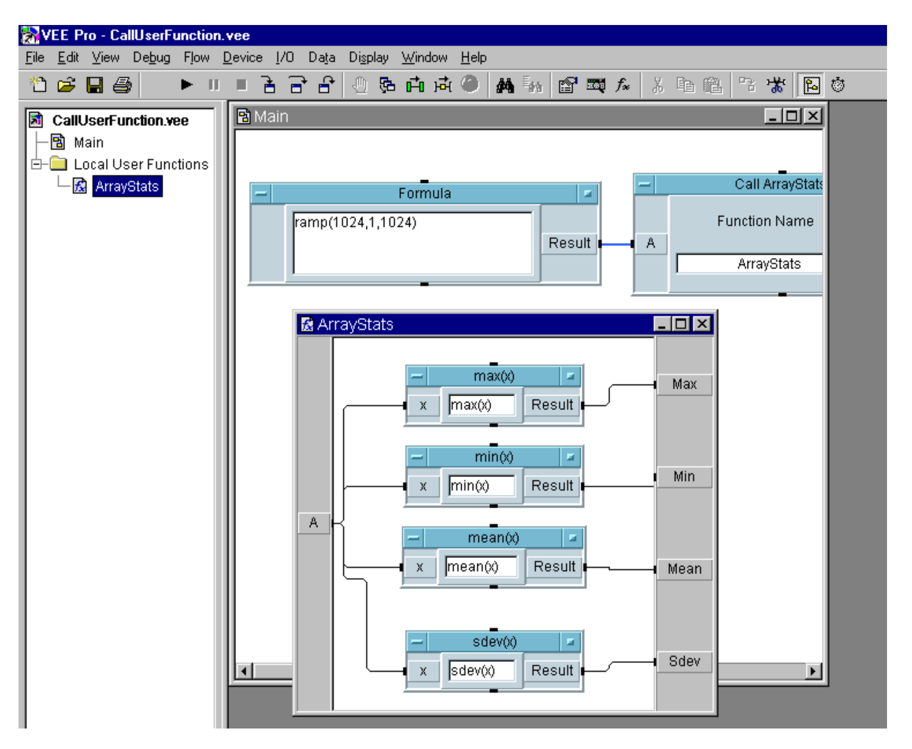


Abbildung 8-10. Verwenden des Program Explorer mit UserFunctions

Verwenden von Bibliotheken mit Agilent VEE UserFunctions

Sie können von vorhandenen VEE-Testprogrammen profitieren, indem Sie `UserFunctions` wiederverwenden. Wenn Sie ein Programm speichern, werden die `UserFunctions` automatisch ebenfalls gespeichert. Eine `UserFunction` kann ein VEE-Programm oder eine Bibliothek aus logisch zusammengehörigen `UserFunctions` enthalten.

Es gibt zwei Möglichkeiten, vorhandene `UserFunctions` in ein neues Programm zu bringen:

1. Legen Sie eine Kopie der Original-`UserFunctions` mit dem Befehl `File` ⇒ `Merge Library...` im aktuellen Programm ab (wobei Sie die separate Kopie jeder `UserFunction` beibehalten). Diese zusammengefügten `UserFunctions` können geändert werden; verwenden Sie daher den Befehl `File` ⇒ `Merge Library...`, wenn Sie vorhaben, die `UserFunctions` zu ändern.

-ODER-

2. Rufen Sie die Original-`UserFunctions` mit dem Objekt `Device` ⇒ `Import Library` auf; dieses Objekt greift auf die Originalfunktionen in einer anderen Datei zu, ohne eine Kopie davon zu erstellen. Diese `UserFunctions` werden zur Ausführungszeit importiert. Dadurch wird die Ladezeit besser verteilt sowie Platten- und Speicherplatz gespart. Importierte `UserFunctions` können angezeigt werden (etwa zur Fehlerbehebung), aber nicht geändert. Sie können jedoch die entsprechenden Originaldateien ändern. Sie können auch importierte `UserFunctions` programmgesteuert löschen über das Objekt `Device` ⇒ `Delete Library`.

Verwenden Sie daher *Merge* `UserFunctions`, wenn Sie eine neue Kopie der Funktion zum Ändern eines eigenständigen Programms benötigen oder zum *Importieren* von `UserFunctions`, wenn Sie eine einzelne Quelle für die Funktion verwenden oder Platz sparen wollen.

Übung 8-2: Erstellen und Zusammenführen einer Bibliothek von UserFunctions

In dieser Übung erstellen Sie ein Berichtsgeneratorprogramm, das eine VEE-Bibliothek von `UserFunctions` enthält. Anschließend erstellen Sie ein neues Programm, das die Bibliothek der `UserFunctions` zusammenführt.

Erstellen einer Bibliothek von UserFunctions

1. Erstellen Sie das Programm der obersten Ebene (ohne die Details der `UserFunctions` zu programmieren).

   a. Erstellen Sie vier `UserFunctions`: `BuildRecAry` mit einem Ausgangs-Pin, `ReportHeader`, `ReportBody` mit einem Eingangs-Pin und `ReportDisplay`. Stellen Sie alle `UserFunctions` als Symbol dar.

   b. Erstellen Sie im `Hauptprogramm` vier Objekte `Device ⇒ Call`, konfiguriert und verbunden wie in Abbildung 8-11 gezeigt. Speichern Sie das Programm als `Report.vee`.

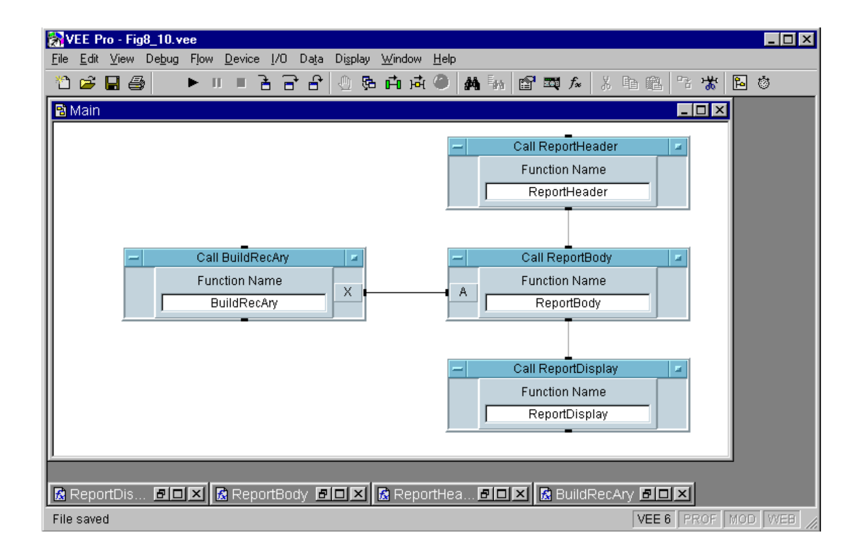


Abbildung 8-11. Report.vee von der obersten Ebene

Hinweis

Das `Call`-Objekt erfordert keine Klammern beim Verweis auf eine `UserFunction`. Wenn Sie die Funktion von einem `Formula`-Objekt aus aufrufen, müssen Sie die Klammern angeben, ob die Funktion Parameter verwendet oder nicht.

Die vier `UserFunctions` bilden eine Bibliothek. Sie können sie auflisten, indem Sie `Edit` ⇒ `Edit UserFunction...` anklicken. Klicken Sie `Cancel` an, um das Listenfeld zu schließen.

2. Programmieren Sie die vier `UserFunctions` wie in den folgenden Abbildungen gezeigt.

   Abbildung 8-12 zeigt die `UserFunction BuildRecAry`. Sie enthält ein `Formula`-Objekt mit dem triadischen Ausdruckstest `if A<=5`. Wenn der Ausdruck `TRUE` ergibt, lautet die Ausgabe des Objekts `"PASS"`. Andernfalls gibt das Objekt `"FAIL"` aus. (Beachten Sie, dass die Klammern erforderlich sind.)

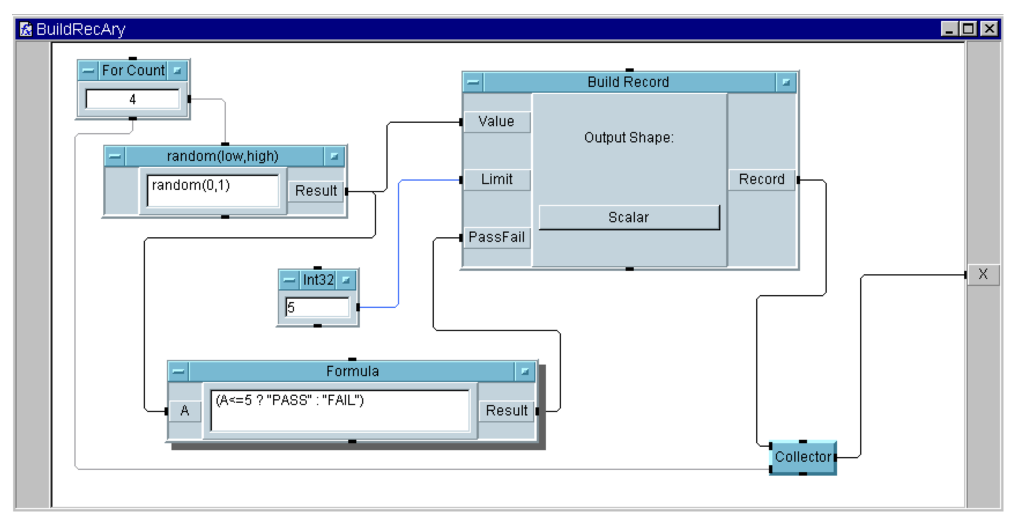


Abbildung 8-12. Die UserFunction "BuildRecAry"

Abbildung 8-13 zeigt die `UserFunction ReportHeader`.

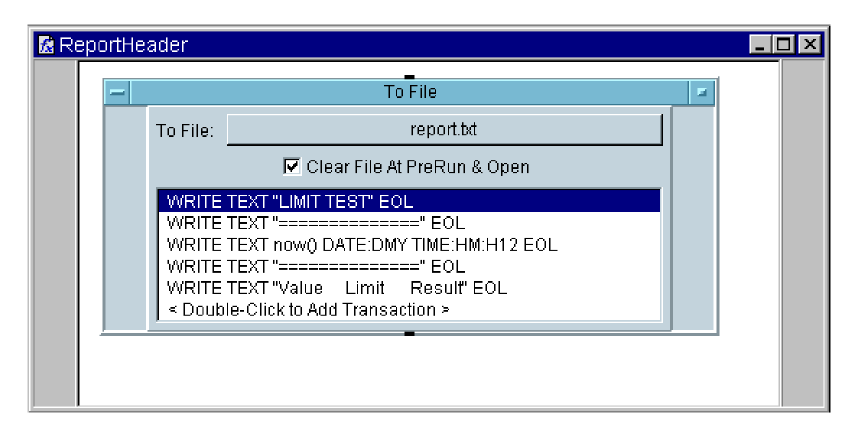


Abbildung 8-13. Die UserFunction "ReportHeader"

Abbildung 8-14 zeigt die `UserFunction ReportBody`. Beachten Sie den Array aus Datensätzen `A[B]`. Wenn sich der `Wert` von `B` ändert von `0` nach `1` nach `2`, können Sie das jeweilige Feld in diesem `Record` einschließlich `Value`, `Limit` und `PassFail` aufrufen über die Notation <*Datensatz*>.<*Feld*>. Beachten Sie, dass die Ausgabe von `For Count` mit Null beginnt. Beachten Sie außerdem, dass für die erste Transaktion `EOL off` gilt.

Der vertikale Balken in Hochkommas, `"|"`, steht für eine konstante Zeichenfolge für einen vertikalen Balken. `FW:5` Steht für die Breite `5` eines Zeichenfolgefelds. `RJ` steht für rechtsbündig.

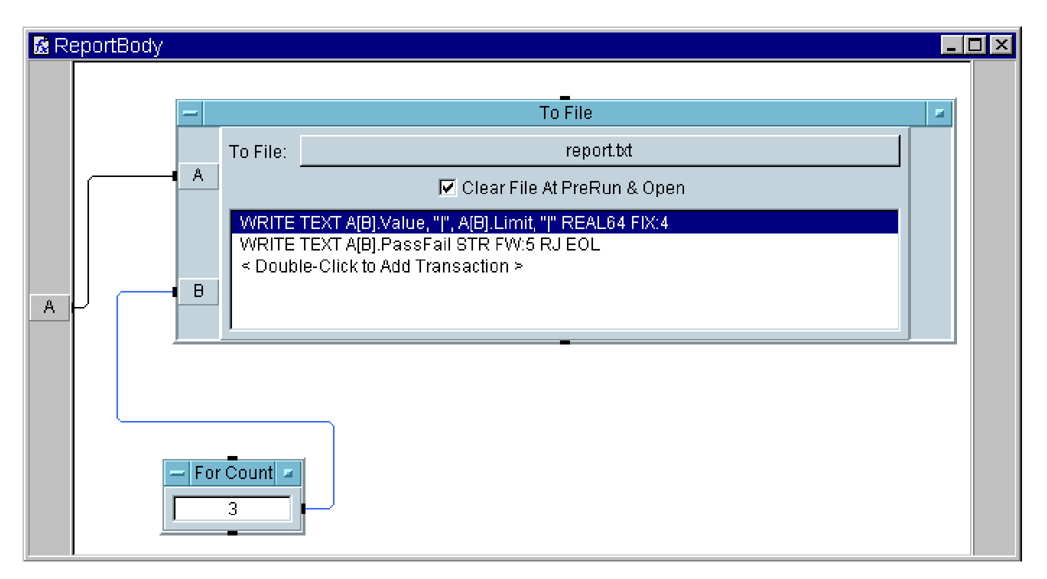


Abbildung 8-14. Die UserFunction "ReportBody"

Abbildung 8-15 zeigt die `UserFunction ReportDisplay` in der Detailansicht. Beachten Sie, dass sie einen Zeichenfolgen-Array bis zum Ende der Datei einliest; dies ist durch das Sternchen (`*`) nach der Formatangabe `STR ARRAY` gekennzeichnet.

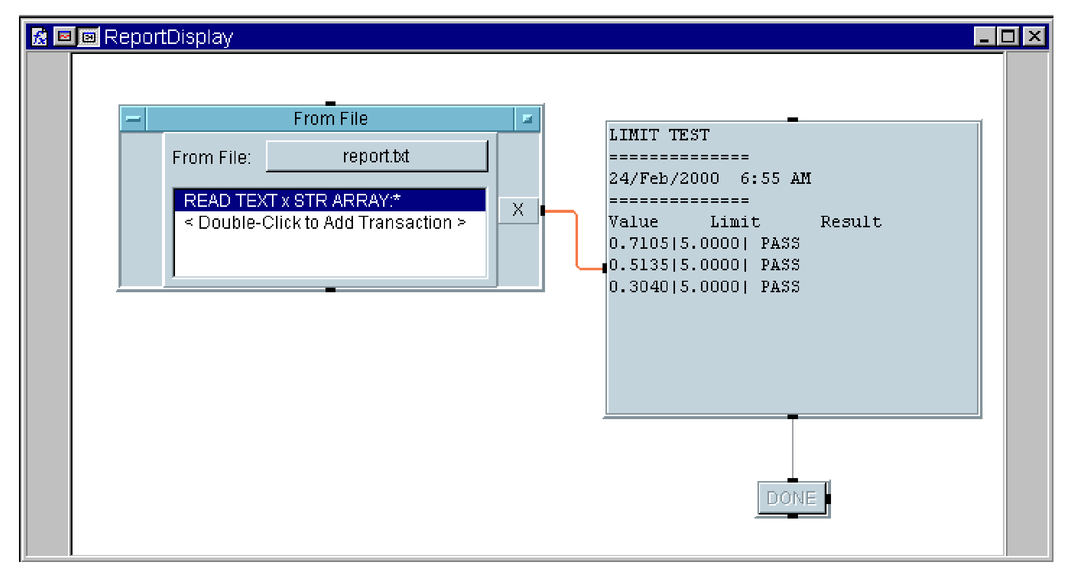


Abbildung 8-15. Die ReportDisplay-Detailansicht

Abbildung 8-16 zeigt die Fensteransicht der `UserFunction ReportDisplay`, wobei `Show Panel on Execute` im Fenster `Properties` ausgewählt ist. Im Feld `Properties` wurde der `Pop-up Panel Title` ebenfalls in `ReportDisplay` geändert. Wählen Sie zum Erstellen des Fensters die Objekte `Confirm (OK)` und `Logging AlphaNumeric` aus und klicken Sie `Edit ⇒ Add to Panel` an. Beachten Sie, dass in der Anzeige `Logging AlphaNumeric` die Auswahl der Option `Show Title Bar` aufgehoben wurde. Beachten Sie außerdem, dass die Schaltfläche `Confirm (OK)` umbenannt wurde in `DONE`. Die Schaltfläche `Confirm (OK)` ist enthalten, um die Anzeige auf dem Bildschirm zu halten, bis der Benutzer sie vollständig gelesen hat.

3. Führen Sie das Programm aus. Das Fenster `ReportDisplay` sollte eingeblendet werden und die Werte wie in Abbildung 8-16 anzeigen.

Klicken Sie DONE an, um die Ausführung abzuschließen. Speichern Sie das Programm anschließend als Report.vee.

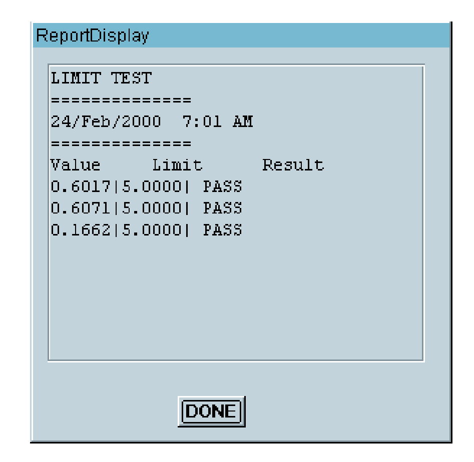


Abbildung 8-16. Die ReportDisplay-Fensteransicht

Erstellen eines weiteren Programms und Einfügen in die Bibliothek

In dieser Übung erstellen Sie ein neues Programm und fügen die Bibliothek in dieses Programm ein. Diese Übung erstellt eine Bibliothek von Funktionen zum Generieren von Berichten. Das neue Programm enthält ein Note Pad-Objekt, das die einzelnen Funktionen in der Bibliothek erläutert. Der Name des Programms lautet RepGen.

Sie können RepGen wiederverwenden, indem Sie neue UserFunctions zur Berichtsgenerierung erstellen, sie in das Programm einfügen und das Objekt Note Pad aktualisieren, um sie zu protokollieren. Anschließend können Sie den Befehl Merge Library... verwenden, um alle Funktionen von RepGen zu nutzen.

1. Wählen Sie File ⇒ New aus.
2. Wählen Sie File ⇒ Merge Library... aus. Wählen Sie Report.vee im Listenfeld Merge Library aus. (Wenn Sie in einem

anderen Verzeichnis arbeiten, geben Sie den vollständigen Pfadnamen an.)

Wählen Sie `Edit ⇒ Edit UserFunction` aus (oder sehen Sie im Program Explorer nach), um sicherzustellen, dass die Bibliothek von `Report.vee` an das neue Programm übertragen wurde. Wenn Sie den Befehl `Merge Library...` verwenden, können Sie zusammengefügte Funktionen genau wie lokale Funktionen verwenden.

3. Wählen Sie `Display ⇒ Note Pad` aus und geben Sie Beschreibungen für die `UserFunction` wie die in Abbildung 8-17 ein. Speichern Sie das Programm anschließend als `RepGen.vee`.

Hinweis

Sie können ein "Programm" aus `UserFunctions` zum Erstellen einer Bibliothek speichern, auch wenn kein tatsächliches VEE-Programm vorliegt, das die Funktionen aufruft.

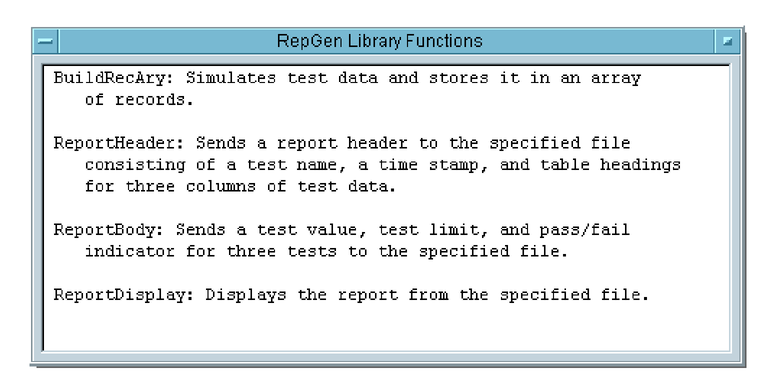


Abbildung 8-17. Die Bibliothek RepGen.vee von UserFunctions

Übung 8-3: Importieren und Löschen von Bibliotheken

Wenn Sie eine Bibliothek von `UserFunctions` erstellt haben, wollen Sie sie wahrscheinlich nicht in jedes Programm einbinden. Sie können die Bibliothek zur Ausführungszeit einbinden, einige der Funktionen verwenden und anschließend die Bibliothek löschen, um Speicherplatz zu sparen. Die Objekte `Import Library` und `Delete Library` wurden für diese Situation konzipiert.

In dieser Übung importieren Sie Funktionen aus dem Programm `RepGen` . Anschließend rufen Sie die Funktion `BuildRecAry` auf, um einige Testdaten zu simulieren und anzuzeigen; anschließend löschen Sie die Bibliothek, um Speicherplatz und Auslagerungsspeicher (Swap Space) freizugeben.

1. Wählen Sie `File ⇒ New` aus.

2. Wählen Sie `Device ⇒ Import Library` aus und platzieren Sie dieses Objekt im `Hauptprogramm`. Stellen Sie die Felder im Objekt `Import Library` wie folgt ein:

Library Type (Typ der Bibliothek)	Das Menü im Feld `Library Type` ermöglicht die Auswahl einer `UserFunction`, einer kompilierten Funktion oder einer Fernfunktion. In diesem Fall wollen Sie mit einer UserFunction-Bibliothek arbeiten; behalten Sie daher den Standardwert bei.
Library Name (Name der Bibliothek)	Der `Library Name` zeigt `myLib` als Standardwert an. Dieser Name wird vom VEE-Programm als Identifikation verwendet, über den die verschiedenen importierten Bibliotheken unterschieden werden können. Das Objekt `Delete Library` verwendet diesen Namen zur Identifikation der zu löschenden Bibliothek. Sie können den Standardnamen verwenden.

File name (Dateiname) Das Feld `File Name` zeigt standardmäßig auf einem PC ein Dialogfenster für das Verzeichnis der Benutzerprogramme an. (HP-UX-Systeme rufen das Verzeichnis auf, von dem aus Sie VEE gestartet hatten.) Geben Sie die Datei an, die die Bibliothek der Funktionen enthält.

Klicken Sie den Standardnamen `myFile` an, um das Listenfeld aufzurufen. Wählen Sie `RepGen.vee` (aus “Erstellen und Zusammenführen einer Bibliothek von UserFunctions” auf Seite 326) aus. Diese Datei befindet sich in dem Verzeichnis, das Sie bei der Installation für Programme angegeben haben.

3. Öffnen Sie das Objektmenü und wählen Sie anschließend `Load Lib` aus, um die Bibliothek sofort zu importieren, statt bis zur Ausführungszeit zu warten. Dieser Befehl ist der Entwicklungsphase sehr nützlich. (Im nächsten Schritt werden Sie feststellen, dass bei der Auswahl von `Select Function` im Objekt `Call` die Funktionen mit der Bibliotheks-Identifikation zuerst gekennzeichnet sind, z. B. `myLib.BuildRecAry`.)

   Verwenden Sie zum Anzeigen der Bibliothek der importierten Funktionen den `Program Explorer`.

Hinweis Da Sie die Bibliothek importiert haben, können Sie die `UserFunction` nur anzeigen und Unterbrechungspunkte dafür festlegen. Sie können die `User Function` nicht ändern. Verwenden Sie den Befehl `Merge Library...`, um eine `UserFunction` einem Programm hinzuzufügen, das geändert werden kann.

4. Wählen Sie `Device` ⇒ `Call` aus und platzieren Sie dieses Objekt unter dem Objekt `Import Library`. Verbinden Sie den Sequenzausgangs-Pin von `Import Library` mit dem Sequenzeingang am Objekt `Call`.

5. Öffnen Sie das `Call Function`-Objektmenü und klicken Sie `Select Function` an, um eine Liste der mit dem Befehl `Load Lib` importierten Funktionen anzuzeigen. Wählen Sie `myLib.BuildRecAry` aus (siehe Abbildung 8-18).

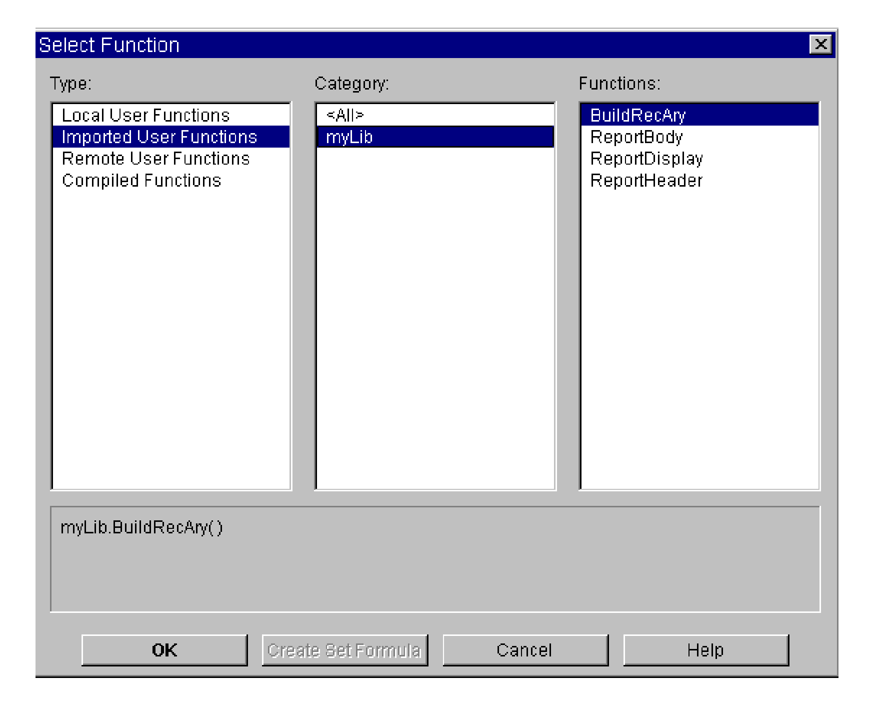


Abbildung 8-18. Auswählen einer Funktion aus einer importierten Bibliothek

VEE fügt die Funktion automatisch im Feld `Function Name` ein und fügt den erforderlichen Ausgangsanschluss hinzu. Sie könnten auch `myLib.BuildRecAry` im Feld `Function Name` eingeben, um dieselben Ergebnisse zu erzielen. Verwenden Sie `Select Function`, wenn Sie die Namen der Funktionen in der Bibliothek auflisten wollen.

6. Wählen Sie eine `AlphaNumeric`-Anzeige aus, vergrößern Sie sie und verbinden Sie sie mit dem Datenausgang von `Call`.

7. Wählen Sie `Device` ⇒ `Delete Library` aus und platzieren Sie dieses Objekt unter dem Objekt `Call`. Verbinden Sie die Sequenz-Pins, sodass die Bibliothek nach dem Aufruf der Funktion `BuildRecAry` gelöscht wird. Sie können den Standardwert für `Library Name` beibehalten, da dies der gleiche Name ist, den Sie mit dem Objekt `Import Library` verwendet hatten.

8. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 8-19. Speichern Sie das Programm als `libraries.vee`.

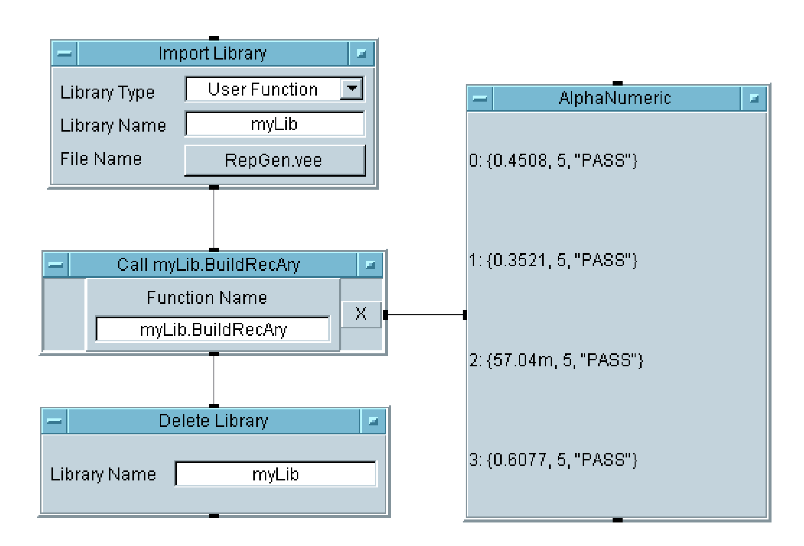


Abbildung 8-19. Aufrufen einer Funktion aus einer Bibliothek

Beachten Sie folgende Punkte bei den Namen zusammengefügter oder importierter Funktionen:

- Wenn eine zusammengefügte Funktion den gleichen Namen hat wie eine lokale Funktion, zeigt VEE einen Fehler an.

- Eine importierte Funktion darf zwar den gleichen Namen haben wie eine lokale Funktion; wenn jedoch nur der Funktionsname verwendet wird, wird die lokale Funktion aufgerufen. Sie können die importierte Funktion mit der Syntax `MyLib_func()` explizit aufrufen, wie im Objekt `Call` in Abbildung 8-19.

- Wenn zwei importierte Bibliotheken identische Funktionsnamen enthalten, ist das Ergebnis nicht bestimmt. Beachten Sie, dass das Objekt `Call` den Namen der Bibliothek `myLib.BuildRecAry` verwendet, somit ist eine Verwechslung ausgeschlossen. Auch wenn es eine andere lokale oder importierte Funktion mit diesem Namen gibt, ist `BuildRecAry` durch die Angabe von Namen und Position eindeutig.

Finden von Funktionen in großen Programmen

VEE bietet eine Funktion `Find` im Menü `Edit` zum Suchen von Objekten und Text in großen Programmen. Öffnen Sie beispielsweise das Programm `Solitaire.vee` im Verzeichnis `Examples/Games`. Wechseln Sie zur `Detailansicht` und klicken Sie `Edit ⇒ Find...` an, um das in Abbildung 8-20 dargestellte Dialogfenster anzuzeigen.

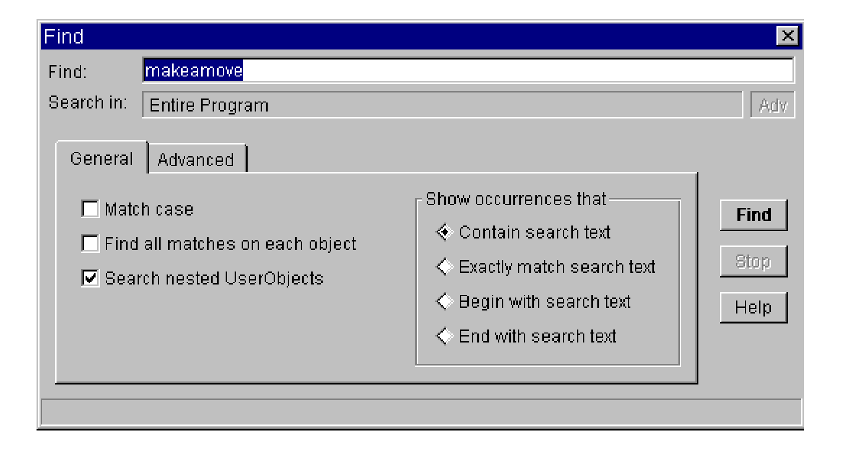


Abbildung 8-20. Das Dialogfenster "Find"

Geben Sie `makeamove` (eine `UserFunction` in diesem Programm) wie in der Abbildung gezeigt ein und klicken Sie `Find` an. VEE findet automatisch die `UserFunction` mit dem Namen `makeamove` und zeigt den Teil des Programms an, von dem aus sie aufgerufen wurde (siehe Abbildung 8-21).

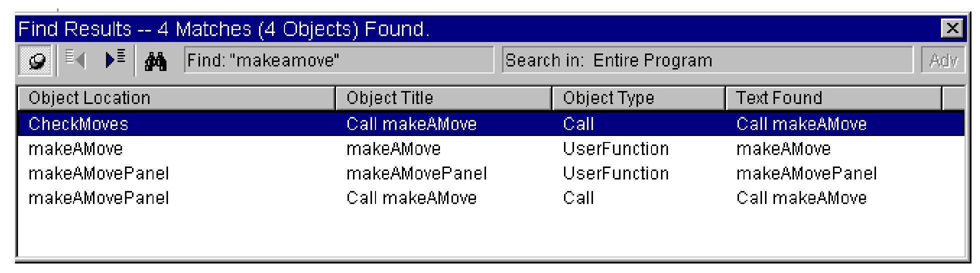


Abbildung 8-21. Das Dialogfenster "Find Results"

Sie können mit `Find` to ein beliebiges Objekt oder einen Text wie beispielsweise Variablen, Titel, Einstellungen von Objekten etc. finden. Klicken Sie doppelt auf eine beliebige Linie im Fenster `Find Results`, um ein Objekt zu suchen.

Hinweis

Sie können `Find` auch verwenden, indem Sie den Mauszeiger auf ein Objekt im `Program Explorer` positionieren und mit der rechten Maustaste klicken. Dadurch wird der Umfang der Suche auf die bestimmte `UserFunction` bzw. das `UserObject` eingegrenzt.

Zusammenfügen von Agilent VEE-Programmen

Die einfachste Möglichkeit, vorhandene Programme weiterzuverwenden, ist das Zusammenfügen eines älteren Programms mit dem aktuellen Test. Sie können Programme weiterverwenden, indem Sie sie zusammenfügen und entsprechend Ihren Anforderungen anpassen.

Der Befehl `File` ⇒ `Merge...` fügt den Inhalt eines Programms oder einer Gruppe gespeicherter Objekte in den Arbeitsbereich ein, wobei der vorhandene Inhalt des Arbeitsbereichs erhalten bleibt. Standardmäßig zeigt `File` ⇒ `Merge...` ein Verzeichnis von Programmen an, die zusammen mit VEE ausgeliefert werden. Hierzu gehören allgemeine Programme wie beispielsweise die Anzeige eines Balkendiagramms und ein Tastenblock für die Benutzereingabe (z. B. für Kennungen).

Übung 8-4: Zusammenfügen eines Programms zur Anzeige von Balkendiagrammen

In dieser Übung fügen Sie ein bestehendes Program mit einem neuen Programm zusammen. Das Beispiel verwendet ein Programm aus dem Verzeichnis VEE `Lib`; Sie können jedoch jedes beliebige Programm verwenden. Sie erstellen einen Array aus fünf Werten von 1 bis 5 mit der Funktion `ramp()`. Statt den Array mit einer der internen Anzeigen von VEE darzustellen, fügen Sie das Programm `BarChart` in das aktuell erstellte Programm ein.

1. Wählen Sie ⇒ `Formula` aus und platzieren Sie dieses Objekt im linken Arbeitsbereich.

2. Löschen Sie den Dateneingangsanschluss.

3. Ändern Sie die Standardformel in `ramp(5,1,5)`.

   Der erste Parameter ist die gewünschte Anzahl der Elemente im ramp-Array. Der zweite Parameter gibt die erste Zahl an, und der dritte Parameter ist die letzte Zahl. Weitere Informationen zu dieser Funktion können Sie anzeigen, indem Sie `Help` im `Formula`-Objektmenü auswählen,

nachdem die Formel jetzt den ramp-Aufruf enthält. (Oder rufen Sie Help ⇒ Contents & Index auf und verwenden Sie die Funktion Search im Ordner Index .)

4. Klicken Sie File ⇒ Merge... an, um das Listenfenster Merge File aufzurufen. Wählen Sie BarChart.vee und platzieren Sie das Objekt rechts von dem Objekt Formula. Verbinden Sie die beiden Objekte.

5. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 8-22.

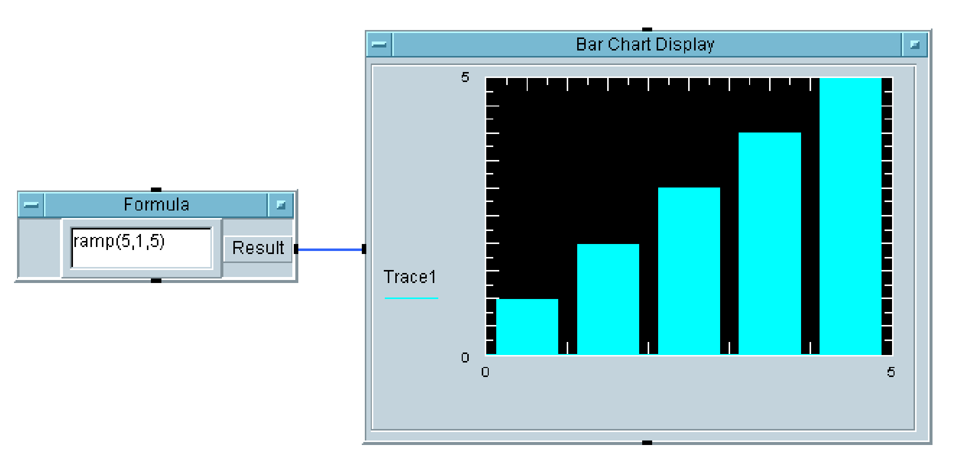


Abbildung 8-22. Zusammenfügen des Balkendiagramm-Programms

Beachten Sie, dass die Anzeige des Balkendiagramms einen eindimensionalen Array erfordert und die Werte als vertikale Balken darstellt. Sie verwendet die erforderliche Anzahl von Balken zum Anzeigen der Werte in dem Array. Öffnen Sie die Detailansicht der Anzeige, um zu sehen, wie das Programm erstellt wird. Sie können Beispiele in der Bibliothek anzeigen, um weitere Anregungen zu Programmen zu erhalten.

Hinweis

Der Befehl File ⇒ Merge wird zum Zusammenfügen von UserObjects und Objekten verwendet. Der Befehl File ⇒ Merge Library wird zum Zusammenfügen von UserFunctions verwendet.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Definieren einer `UserFunction` und Vergleichen mit einer kompilierten Funktion und einer Fernfunktion.
- Erstellen, Aufrufen und Bearbeiten einer `UserFunction`.
- Erstellen, Zusammenfügen, Importieren und Löschen von `UserFunction`-Bibliotheken.
- Verwenden der `Find`-Funktion in einem der Spielprogramme.
- Zusammenfügen eines vollständigen VEE-Programms mit dem aktuellen Programm.

9

Sequenzieren von Tests

Sequenzieren von Tests

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Das Objekt `Sequencer`
- Konfigurieren eines Tests für den `Sequencer`
- Erstellen einer Reihenfolge für die Testausführung entsprechend den Ergebnissen zur Ausführungszeit
- Aufrufen der vom `Sequencer` protokollierten Daten
- Möglichkeiten zum Übergeben von Daten an oder von `Sequencer-Tests`
- Ausführen einer Analyse mit protokollierten Daten vom `Sequencer`
- Speichern von `Sequencer`-Testdaten

Erforderliche Zeit für dieses Kapitel: 2 Stunden

Überblick

In diesem Kapitel lernen Sie die Grundfunktionen zur Verwendung des Objekts `Sequencer` kennen. Das `Sequencer`-Objekt kann eine Reihe von Sequenz-Transaktionen ausführen, die wiederum eine `UserFunction`, eine kompilierte Funktion oder eine Fernfunktion ausführen können. Normalerweise wird der `Sequencer` zur Ausführung einer Reihe von Tests verwendet.

Einige der Vorteile des `Sequencer` sind:

- Einfache Entwicklung eines Testplans
- Breites Spektrum von Verzweigungsfunktionen zwischen Tests
- Wichtige Komponente zum Aufbau einer angepassten Testausführung
- Möglichkeit zum Aufruf von Tests in VEE und anderen Sprachen
- Automatische Protokollierung der Testergebnisse

Hinweis Der `Sequencer` ist eine der leistungsstärksten Funktionen von VEE. Weitere Informationen zum `Sequencer` finden Sie in der Online-`Hilfe`.

Die erste Übung zeigt, wie Sie einen Test für das `Sequencer`-Objekt konfigurieren, einen Test im Ausführungsfluss hinzufügen, einfügen oder löschen und auf die Testdaten zugreifen können, die vom `Sequencer` protokolliert wurden. Die Übung simuliert die Testergebnisse mit der Funktion `random()`.

In der zweiten Übung lernen Sie, wie Sie Daten strukturieren, die mit globalen Variablen an Tests übergeben wurden, wie Sie `UserFunctions` über den `Sequencer` aufrufen und wie Sie `Sequencer`-Dateien in Dateien protokollieren. Als letztes analysieren Sie Teile der Daten.

Hinweis Informationen zur Verwendung eines Statusfensters zur Aktualisierung einer Testsequenz finden Sie im Abschnitt “Erstellen einer Statusanzeige” auf Seite 420.

Hinweis

Zusätzlich zu den Übungen in diesem Kapitel können Sie weitere praktische Erfahrungen mit dem `Sequencer` sammeln mit der Übung in “Sequenzieren von Tests” auf Seite 540 in Anhang A, “Zusätzliche Übungen,”

Verwenden des Sequencer-Objekts

Das `Sequencer`-Objekt führt Tests in einer angegebenen Reihenfolge entsprechend den Ergebnissen zur Ausführungszeit aus. Die einzelnen Tests können VEE `UserFunctions`, kompilierte Funktionen, Fernfunktion oder andere Ausdrücke sein, die einzelne Ergebnisse zurückgeben. Dieses Ergebnis wird mit der Testspezifikation verglichen, um festzustellen, ob der Test erfolgreich war oder nicht. Der `Sequencer` verwendet dann ein Kennzeichen "erfolgreich" oder "fehlgeschlagen", um zu ermitteln, welcher Test als nächstes ausgeführt werden soll.

Es gibt sechs verschiedene Optionen zum Verzweigen zum nächsten Test. Diese Optionen umfassen die Ausführung des nächsten Tests, die Wiederholung des letzten Tests oder den Rücksprung zu einem früheren Test. Übung 9-1 erläutert diese Verzweigungsoptionen ausführlich. Der `Sequencer` kann den Benutzer auch zu einer Eingabe auffordern, um zu entscheiden, welche Aktion ausgeführt werden soll.

Nach der Ausführung der angegebenen Tests protokolliert der `Sequencer` die Testdaten automatisch an einen Ausgangsanschluss. Von dieser Stelle aus können Daten analysiert, angezeigt oder zur späteren Überprüfung in einer Datei gespeichert werden.

Erstellen einer Reihenfolge für die Ausführung eines Tests

In dieser Übung simulieren Sie Testergebnisse mit der Funktion `random()`, erstellen eine Reihenfolge für die Ausführung der Tests, erfahren mehr über die Änderung dieser Reihenfolge und rufen spezifische Daten aus den protokollierten Ergebnissen ab.

Übung 9-1: Konfigurieren eines Tests

Hinweis

Das Beispiel erläutert, wie Sie die Funktion `random()` mit einem bestimmten Bereich erwarteter Testergebnisse implementieren; Sie können dasselbe Prinzip jedoch auch beim Konfigurieren anderer Tests anwenden.

1. Wählen Sie `Device` ⇒ `Sequencer` aus und platzieren Sie dieses Objekt oben links im Arbeitsbereich.

2. Wählen Sie `Display` ⇒ `AlphaNumeric` aus, platzieren Sie diese Anzeige unter dem `Sequencer`, verbreitern Sie sie und verbinden Sie den Ausgangsanschluss `Log` des `Sequencer` mit dem Dateneingang von `Alphanumeric`.

3. Klicken Sie doppelt auf die Transaktionsleiste `Sequencer`, um das Dialogfenster `Sequence Transaction` aufzurufen. Stellen Sie die Felder wie folgt ein:

Hinweis

Denken Sie beim Ändern der Felder daran, dass Sie zum Ändern neuer Felder diese anklicken oder mit der **Tabulatortaste** in das entsprechende Feld springen können. Verwenden Sie die Kombination **Umschalttaste-Tabulatortaste**, um den Cursor ein Feld rückwärts zu verschieben. Drücken Sie die **Eingabetaste** nur, wenn Sie mit dem Bearbeiten des Dialogfensters fertig sind.

TEST: Der Standardname lautet `test1`; Sie können diesen Namen verwenden. Dies ist nur die Beschriftung für den Test im `Sequencer`, nicht der Name der Funktion selbst.

SPEC NOMINAL: Gibt den erwarteten Testwert an. Der Standardwert ist `.5`. Ändern Sie diesen Wert in `.25`, und ändern Sie die Obergrenze im Feld `RANGE` (ganz rechts) von `1` in `.5`.

FUNCTION: Der Standardeintrag `testFunc(a)` enthält eine Funktion zur Ausführung des Testes. Ersetzen Sie in diesem Fall das Standardfeld durch die Funktion `random()`. `Random()` gibt einen `Real64`-Wert zwischen `0` und `1` zurück, der das Testergebnis simuliert. Dieses Ergebnis wird mit der Spezifikation des Tests verglichen.

Das Objekt `random(low,high)` befindet sich in der Kategorie `Probability & Statistics` im Fenster `Function & Object Browser`. Denken Sie daran, dass Sie diese mathematische Funktion von einem beliebigen Ausdrucksfeld aus mit einem `Formula`-Objekt aufrufen können. Wenn Sie die Parameter low und high nicht wie in diesem Beispiel gezeigt angeben, verwendet die Funktion die Standardparameter `0` und `1`.

Sie können die weiteren Standardwerte beibehalten. Diese Konfiguration liefert in ca. der Hälfte aller Fälle das Ergebnis `PASS` (erfolgreich). Das Dialogfenster sollte ungefähr aussehen wie in Abbildung 9-1.

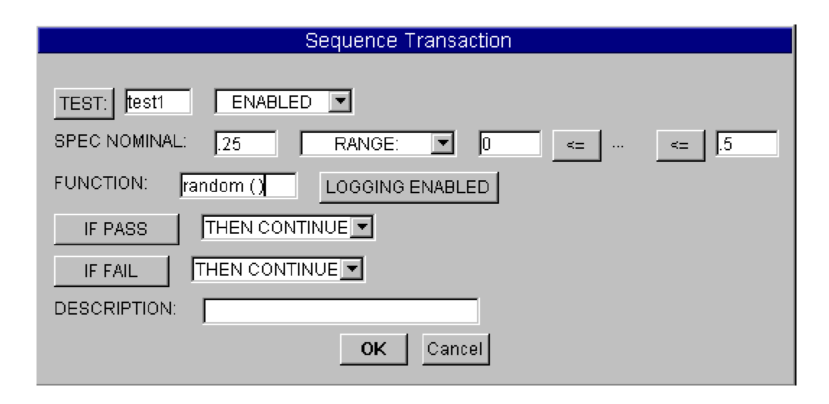


Abbildung 9-1. Das Dialogfenster "Sequence Transaction"

Klicken Sie `OK` an, um das Dialogfenster zu schließen. Sie sehen die Transaktion `test1 0 <= (.25) <= .5` in der ersten Transaktionsleiste. Dies bedeutet, dass `test1` erfolgreich ist, wenn der zurückgegebene Wert im Bereich zwischen `0` und `.5` (je einschließlich) liegt. Das erwartete Ergebnis liegt bei ca. `.25`.

4. Führen Sie das Programm aus. Es sollten den Namen des Tests anzeigen, das Testergebnis und das Kennzeichen Pass/Fail (erfolgreich/fehlgeschlagen) (`1` für `PASS`, `0` für `FAIL`); siehe Abbildung 9-2.

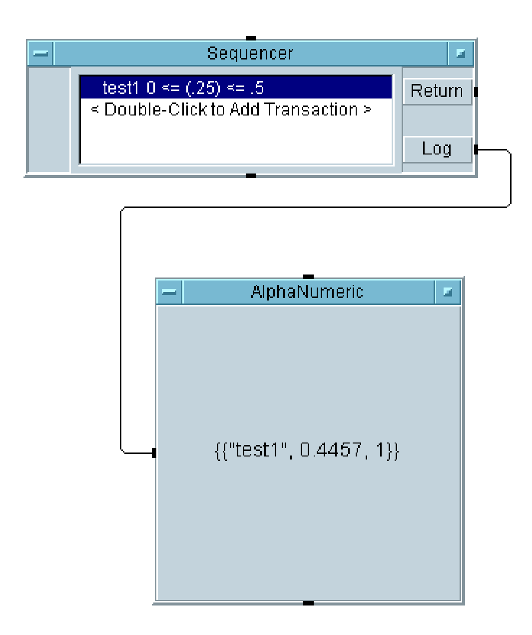


Abbildung 9-2. Konfigurieren eines Tests

Vor der Fortsetzung sehen Sie sich Tabelle 9-1 an, um sich einen Überblick über die verschiedenen Auswahlmöglichkeiten im Dialogfenster `Sequence Transaction` zu verschaffen. Öffnen Sie das Dialogfenster erneut, indem Sie doppelt auf die Transaktionsleiste klicken. Öffnen Sie die verschiedenen Menüs und lesen Sie über die verschiedenen Möglichkeiten nach.

Tabelle 9-1. Dialogfenster "Sequence Transaction"

Sequenz-Transaktionsfeld	Erläuterung
TEST:	Eindeutiger Name für den Verweis auf den Test im `Sequencer`. Die Standardnamen beginnen mit `test1` und werden mit jedem Test hochgezählt. Die Auswahl von `TEST:` bedeutet, dass ein Testergebnis mit den Spezifikationen verglichen wird, und es erfolgt entsprechend der Konfiguration eine Verzweigung auf den nächsten Test. Die Schaltfläche `TEST:` schaltet um auf `EXEC:`. Wenn Sie das Feld `TEST:` auf `EXEC:` umschalten, wird der Test ohne Vergleich zwischen einem Testergebnis und den Spezifikationen ausgeführt. Sie könnten beispielsweise `EXEC:` auswählen, wenn die `UserFunction` globale Variablen einrichtet. Mit der Auswahl `EXEC:` wird außerdem die Protokollierung für den Test ausgeschaltet.
ENABLED	Legt fest, wann ein Test ausgeführt werden soll. Dieses Menü zeigt vier Auswahlmöglichkeiten an: ■ `ENABLED` führt den Test unter allen Umständen aus. ■ `ENABLED IF:` führt den Test aus, wenn der angegebene Ausdruck `TRUE` ist. Der Test kann beispielsweise aktiviert sein, wenn der Eingangs-Pin `A` den Wert `1 (A == 1)` enthält. Sie können `ENABLED IF:` zur Testüberwachung verwenden. Auf diese Weise kann beispielsweise ein bestimmter Test immer nach zehn Durchläufen gestartet werden. ■ `DISABLED` ist das Gegenteil von `ENABLED`. ■ `DISABLED IF:` ist das Gegenteil von `ENABLED IF:`.

Tabelle 9-1. Dialogfenster "Sequence Transaction" (Fortsetzung)

Sequenz-Trans-aktionsfeld	Erläuterung
SPEC NOMINAL:	Der von dem Test erwartete Wert.
RANGE:	Gibt den Bereich der Testwerte an. Dieses Menü zeigt vier Auswahlmöglichkeiten an: ■ `RANGE` gibt den Bereich der Testwerte an, die eine Bedingung `PASS` kennzeichnen. Sie können außerdem unter den üblichen Vergleichen wählen: `>`, `>=`, `<`, `<=`, `==`, `!=`. ■ `LIMIT` verwendet nur einen einzigen Wert für den Vergleich von Testdaten. ■ `TOLERANCE` gibt den Bereich der "erfolgreichen" Werte durch Addieren oder Subtrahieren der angegebenen Toleranz zu bzw. von dem Wert `SPEC NOMINAL` an. ■ `%TOLERANCE` gibt den Bereich der "erfolgreichen" Werte durch Addieren oder Subtrahieren einer prozentualen Toleranz des Werts `SPEC NOMINAL` zu bzw. von der normalen Spezifikation an.
FUNCTION:	Gibt den auszuführenden Test an. Sie können `UserFunctions`, kompilierte Funktionen oder Fernfunktionen aufrufen, oder Sie können einen Ausdruck eintragen, der ausgewertet werden soll. Das Ergebnis der aufgerufenen Funktion (bzw. des ausgewerteten Ausdrucks) wird mit den Spezifikationen verglichen. Wenn eine `UserFunction` mehr als einen Wert zurückgibt, geht VEE davon aus, dass der oberste Ausgabe-Pin das zu testende Ergebnis enthält. Funktionen können auch kombiniert und verschachtelt werden: Beispiel: `(random(0,myfunc()+3,100)*2)`.

Tabelle 9-1. Dialogfenster "Sequence Transaction" (Fortsetzung)

Sequenz-Trans-aktionsfeld	Erläuterung
LOGGING ENABLED	Protokolliert Testdaten. Zum Angeben von Protokollierungsoptionen öffnen Sie das `Sequencer`-Objektmenü, wählen `Properties` aus, klicken den Ordner `Logging` an und treffen Ihre Auswahl in der Liste. Standardmäßig sind `Name`, `Result` und `Pass` ausgewählt. Außerdem steht ein Feld zur Verfügung, in dem Sie zwischen `Log to Output Pin Only` (Nur an Ausgangs-Pin protokollieren) und `Log Each Transaction To:` (Jede Transaktion protokollieren an:) auswählen können. Wenn die Protokollierung aktiviert ist, protokolliert jeder Test einen Datensatz. Schaltet um auf `LOGGING DISABLED`.
IF PASS	Legt die Verzweigungsanleitungen fest. Wenn der Test erfolgreich war, springt VEE in diese Zeile, um die Verzweigungsanweisungen auszuführen. `IF PASS` weist VEE an, entsprechend der Auswahl im Dropdown-Menü zu verzweigen. Diese Schaltfläche schaltet außerdem um zu `IF PASS CALL:`. `IF PASS CALL:` weist VEE an, die angegebene Funktion aufzurufen und anschließend zur Verzweigungs-Menüauswahl zu springen. (Schlagen Sie auch unter `THEN CONTINUE`, dem nächsten Element in dieser Tabelle, nach.)

Tabelle 9-1. Dialogfenster "Sequence Transaction" (Fortsetzung)

Sequenz-Trans-aktionsfeld	Erläuterung
THEN CONTINUE	Legt die Testverzweigung fest. Das Dropdown-Menü `THEN CONTINUE` (für `IF PASS` und `IF FAIL`) enthält sechs Verzweigungsoptionen: ■ `THEN CONTINUE` führt den nächsten im `Sequencer` konfigurierten Test aus. ■ `THEN RETURN:` weist VEE an, die Ausführung von Tests zu stoppen und den angegebenen Ausdruck am `Return`-Ausgangs-Pin des `Sequencer` anzulegen. ■ `THEN GOTO:` springt zu dem in diesem Feld angegebenen Test. ■ `THEN REPEAT` wiederholt den aktuellen Test bis zu der im Feld `MAX TIMES:` angegebenen Anzahl. Wenn die Bedingung `PASS/FAIL` auch nach der maximalen Anzahl von Wiederholungen weiterhin erfüllt ist, fährt VEE mit dem nächsten Test fort. ■ `THEN ERROR:` stoppt die Ausführung durch Generieren einer Fehlerbedingung mit der angegebenen Fehlernummer. Ein Fehler kann mit dem Ausgangs-Pin `Error` des `Sequencer` erfasst werden. Keiner der anderen Ausgangs-Pins sendet Daten. ■ `THEN EVALUATE:` ruft die angegebene `User Function` auf, die eine Zeichenfolge zurückgeben muss, die wiederum eine Verzweigungs-Menüoption angeben muss. Gültige Zeichenfolgeergebnisse von der `UserFunction` sind: `"Continue"`, `"Return` *<expr>*`"`, `"Goto` *<name>*`"`, `"Repeat` *<expr>*`"`, `"Error <expr>"`, wobei *<expr>* ein beliebiger gültiger VEE-Ausdruck und *<name>* der Name eines Tests in einer Sequenz ist. Diese Option gibt Ihnen die Möglichkeit, den Benutzer zu fragen, welche Aktion als nächste ausgeführt werden soll.

Tabelle 9-1. Dialogfenster "Sequence Transaction" (Fortsetzung)

Sequenz-Trans-aktionsfeld	Erläuterung
IF FAIL	Verzweigungsanleitungen. Wenn der Test fehlgeschlagen ist, springt VEE in diese Zeile, um die Verzweigungsanweisungen auszuführen. `IF FAIL` schaltet um auf `IF FAIL CALL:`. Die Optionen sind die gleichen wie für `IF PASS`.
DESCRIPTION:	Textkommentare zu dem Test. Diese Kommentare werden in der `Sequencer`-Transaktionsleiste angezeigt. Sie können zusammen mit dem Testdatensatz gespeichert werden; verwenden Sie hierzu den Ordner `Logging` im Dialogfenster `Properties`.

Hinzufügen, Einfügen oder Löschen eines Tests

In diesem Abschnitt fügen Sie dem `Sequencer`-Objekt eine weitere Test-Transaktion hinzu. Sie können dieselbe `random()`-Funktion zum Simulieren eines Testergebnisses verwenden; dieses Mal vergleichen Sie jedoch das Testergebnis mit einem Grenzwert statt mit einem Wertebereich.

1. Klicken Sie doppelt unter der ersten Transaktionsleiste `Sequencer`, um das Dialogfenster `Sequence Transaction` aufzurufen. Füllen Sie die Felder wie folgt aus:

test2	Verwenden Sie den Standardwert.
SPEC NOMINAL:	Ändern Sie die Einstellung von `.5` in `.25`.
RANGE:	Wählen Sie im Dropdown-Menü `LIMIT:` aus. Wählen Sie `<` für den Operator aus. Ändern Sie `1` in `.5` für den Grenzwert.
FUNCTION:	Ändern Sie das Feld von `testFunc(a)` in `random()`.

Behalten Sie die weiteren Standardwerte bei und klicken Sie `OK` an, um zum `Sequencer` zurückzukehren.

Hinweis

Sie können auch eine Transaktion nach der hervorgehobenen hinzufügen, indem Sie Add Trans... im Objektmenü auswählen.

Der Sequencer-Testplan sollte jetzt die folgende zweite Transaktion enthalten: test2 (.25) < .5. Fügen Sie jetzt eine Transaktion zwischen diesen beiden Tests ein.

2. Vergewissern Sie sich, dass die zweite Transaktionsleiste hervorgehoben ist. Öffnen Sie anschließend das Objektmenü und wählen Sie Insert Trans... aus. Füllen Sie die Felder wie folgt aus:

   TEST: Ändern Sie das Namensfeld in Insert.

   FUNCTION: Ändern Sie die Angabe in random().

   Klicken Sie OK an. In der zweiten Transaktionsleiste wird jetzt Insert 0 <= (.5) <= 1 angezeigt. Führen Sie das Programm aus, um die drei Datensätze aus den drei Tests anzuzeigen. (Eventuell müssen Sie die Anzeige vergrößern, um alle drei Einträge zu sehen.)

3. Löschen Sie jetzt Insert, indem Sie die Transaktionsleiste Insert anklicken, den Mauszeiger auf die Transaktionsleiste Insert positionieren und **Strg-K** drücken.

Hinweis

Sie können auch die Ziel-Transaktionsleiste anklicken und Cut Trans im Objektmenü auswählen. Sie können auch eine zuvor ausgeschnittene Transaktion einfügen, indem Sie Paste Trans im Objektmenü auswählen (oder den Direktaufruf **Strg-Y** verwenden). Auf ähnliche Weise können Sie eine Transaktion mit der Auswahl Copy Trans kopieren.

4. Führen Sie das Programm aus und notieren Sie die beiden Datensätze aus den beiden Tests. Speichern Sie das Programm als seq1.vee. Das Programm sollte jetzt aussehen wie in Abbildung 9-3.

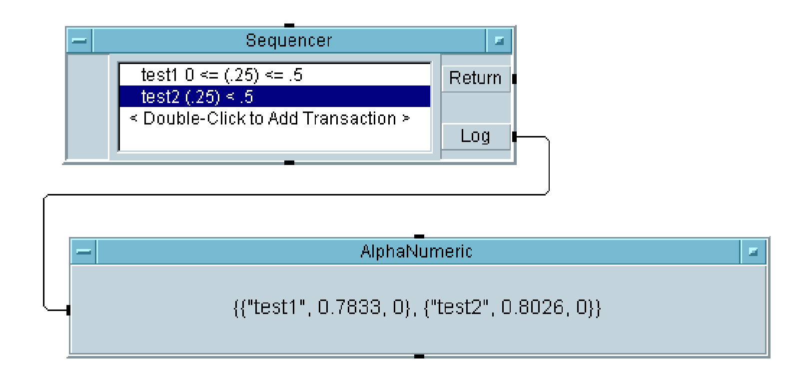


Abbildung 9-3. Ein einfaches Sequencer-Beispiel

Die Klammern kennzeichnen einen Datentyp `Record`. Der `Sequencer` gibt einen `Record of Records` aus, wie in der Anzeige `AlphaNumeric` dargestellt. Dies bedeutet, dass Sie den Sequencer in einer Schleife ausführen und die gleiche Testsequenz mehrmals ausführen können, wobei Sie mehrmals einen Array aus `Records of Records` erhalten.

Aufrufen der protokollierten Testdaten

Der `Sequencer` gibt einen `Record of Records` (Datensatz aus Datensätzen) aus. Jeder Test verwendet den Testnamen als seinen Feldnamen im `Sequencer`-Datensatz. Die Felder in den einzelnen Tests werden entsprechend der Konfiguration für die Protokollierung benannt. Über die Standardkonfiguration mit den Feldern `Name`, `Result` und `Pass` können Sie auf das Ergebnis in `test1` mit der Notation `Log.Test1.Result` zugreifen (siehe Abbildung 9-4).

.

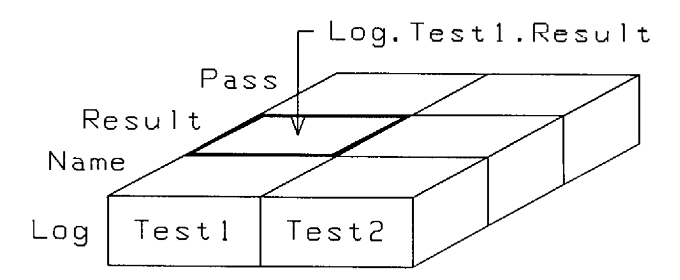


Abbildung 9-4. Ein oder mehrere protokollierte Datensätze

Die folgenden Schritte rufen die Testergebnisse auf.

1. Öffnen Sie `seq1.vee`.

2. Wählen Sie `Device` ⇒ `Formula` aus und platzieren Sie das Objekt unter der Anzeige. Ändern Sie den Ausdruck in `Log.Test1.Result`. (Denken Sie daran, dass in VEE nicht zwischen Groß- und Kleinbuchstaben unterschieden wird. In der Dokumentation wurden bei den Namen nur wegen der besseren Lesbarkeit Großbuchstaben verwendet.)

   Ändern Sie den Namen des Eingangsanschlusses von `A` in `Log`. (Sie können auch den Standardnamen `A` beibehalten; die Formel lautet dann `A.Test1.Result`.) Verbinden Sie den Ausgangsanschluss `Log` des `Sequencer` mit dem Eingangsanschluss `Log` von `Formula`.

3. Wählen Sie `Display` ⇒ `AlphaNumeric` aus und verbinden Sie das Objekt mit dem `Formula`-Ausgang.

4. Führen Sie das Programm aus. Es sollte auf das Feld `Result` in `Test1` zugreifen. Speichern Sie das Programm als `seq2.vee`. Das Programm sollte jetzt aussehen wie in Abbildung 9-5.

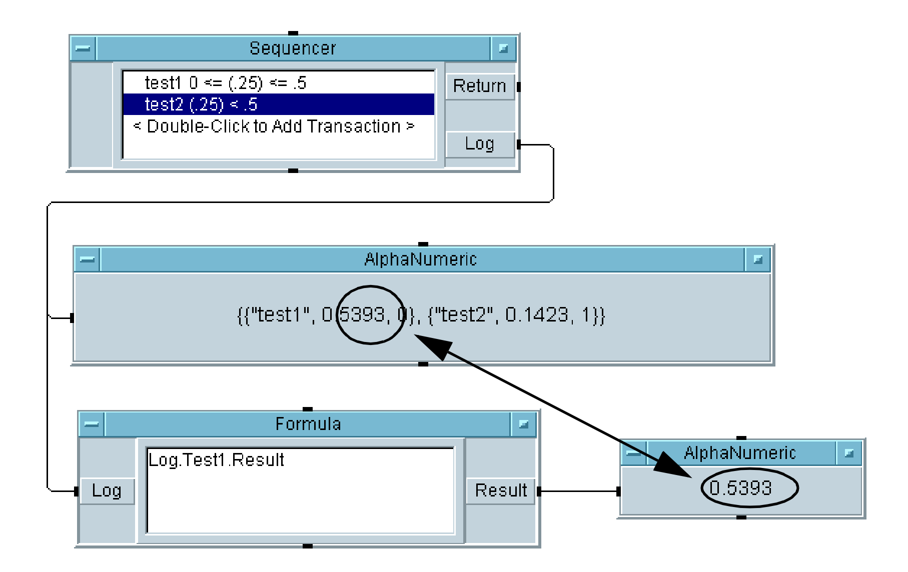


Abbildung 9-5. Aufrufen protokollierter Daten

Hinweis

Jeder Test erstellt bei der Ausführung im `Sequencer` einen Datensatz ("Record") mit dem Namen des Tests. Dieser Datensatz kann in späteren Tests verwendet werden. Sie können beispielsweise `test2` aktivieren, wenn `test1` erfolgreich war (`ENABLED IF: test1.pass == 1`). Wenn Sie die Testdaten in einem Ausdrucksfeld aufrufen müssen, während der Test noch aktiv ist, werden die Testdaten in dem temporären Datensatz `thistest` gespeichert.

5. Ändern Sie die Formel, sodass sie `Log.test1` lautet, und führen Sie das Programm erneut aus. Es sollte den vollständigen Datensatz für `test1` abrufen; dies wird durch die Klammern um die drei Werte in der Anzeige gekennzeichnet.

6. Durch die Änderung der Formel können Sie auf `result`, `pass`, `name` und andere Felder in den Datensätzen `test1` und `test2` zugreifen. Wählen Sie das Register `Logging` im Fenster `Properties` aus und fügen Sie die Felder `Nominal` und `Time Stamp` den protokollierten Datensätzen hinzu. Rufen Sie diese neuen Felder mit dem `Formula`-Objekt auf.

Weitergeben von Daten im Sequencer

In dieser Übung erstellen Sie eine `UserFunction` und rufen Sie aus drei verschiedenen Tests auf. Im ersten Teil geben Sie Daten an die `UserFunctions` über einen Eingangsanschluss am `Sequencer` weiter. Im zweiten Teil ändern Sie das Programm, sodass es eine globale Variable statt eines Eingangsanschlusses verwendet. Dadurch haben Sie die Möglichkeit, eine Funktion im `EXEC`-Modus aufzurufen statt im `TEST`-Modus. Im dritten Teil lernen Sie, wie Sie eine Wellenformausgabe mit einer Maske testen.

Übung 9-2: Weitergeben von Daten mit einem Eingangsanschluss

Führen Sie zunächst die Schritte zum Erstellen der `UserFunction Rand` aus, diese UserFunction simuliert einen Messprozess. `Rand()` fügt dem Ausgang des Objekts `random(low,high)` einen Eingangsparameter hinzu und legt dieses Ergebnis am Ausgangsanschluss ab. `Rand()` wird aus drei verschiedenen Tests aufgerufen.

1. Wählen Sie `Device ⇒ UserFunction` aus. Ändern Sie den Namen von `UserFunction1` in `Rand`.

2. Rufen Sie die Funktion `random(low,high)` ab, löschen Sie die Eingangsanschlüsse, löschen Sie die Parameter und platzieren Sie das Objekt in `Rand`. (Ohne Parameter lauten die Standardwerte `0` und `1`.) Platzieren Sie ein Objekt `A+B` rechts von `random(low,high)`. Verbinden Sie den Ausgang von `random(low,high)` mit dem linken oberen Eingang des Objekts `A+B`.

3. Fügen Sie `Rand` einen Dateneingangsanschluss hinzu. Verbinden Sie den Eingangsanschluss `A` mit dem linken unteren Eingangsanschluss des Objekts `A+B`.

4. Fügen Sie `Rand` einen Datenausgangsanschluss hinzu. Verbinden Sie den Ausgang des Objekts `A+B` mit dem Ausgangsanschluss von `Rand`.

   Die `UserFunction Rand` sollte jetzt aussehen wie in Abbildung 9-6.

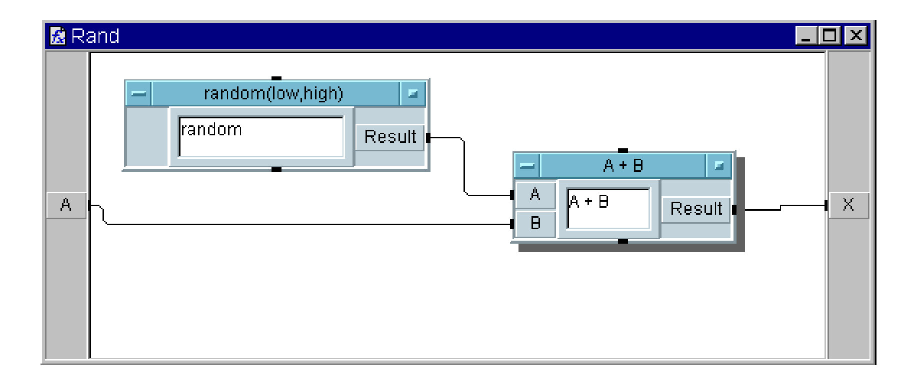


Abbildung 9-6. Die UserFunction Rand

5. Speichern Sie das Programm als `seqdat1.vee`. Schließen Sie das Fenster `Rand` mit der Schaltfläche **x** in der rechten oberen Ecke.

Hinweis

Durch das Schließen des Fensters wird die `UserFunction` nicht entfernt. Sie können dies überprüfen, indem Sie `Edit` ⇒ `Edit UserFunction` anklicken - `Rand` wird in einer Liste der zu ändernden `UserFunctions` aufgeführt. Sie können die Funktion `Rand` auch als Symbol anzeigen; das entsprechende Symbol erscheint am unteren Rand der VEE-Anzeige.

Richten Sie jetzt im `Sequencer` die drei Tests zum Aufruf von `Rand` mit einem `Sequencer`-Eingangs-Pin so ein, dass die Eingangsparameter für `Rand` geliefert werden.

6. Wählen Sie `Device` ⇒ `Sequencer` aus und platzieren Sie dieses Objekt im Hauptprogramm. Fügen Sie dem `Sequencer` einen Eingangsanschluss hinzu. Klicken Sie die Transaktionsleiste an, um das Dialogfenster `Sequence Transaction` aufzurufen. Ändern Sie das Feld `FUNCTION` von `testFunc(a)` in `rand(a)`. Dadurch wird die `UserFunction Rand()` aufgerufen und der Wert an den Eingangsanschluss `A` des `Sequencer` gesendet. Klicken Sie `OK` an, um zur offenen Ansicht des `Sequencer` zurückzukehren.

Hinweis

Sie können auch den Namen eines `Sequencer`-Eingangsanschlusses verwenden, z. B. `A`, um Daten an ein beliebiges Ausdrucksfeld im Fenster `Sequence Transaction` weiterzugeben. Sie können beispielsweise über `A` Daten an `RANGE:` und `SPEC NOMINAL:` weitergeben.

Vergewissern Sie sich, dass die Transaktion hervorgehoben ist, platzieren Sie den Cursor auf der Transaktionsleiste, drücken Sie **Strg-K**, um den Test auszuschneiden, und drücken Sie dann **Strg-Y** drei Mal, um den Test wieder im `Sequencer` einzufügen. (Sie können zum Ausschneiden und Einfügen auch die Objektmenüs verwenden.)

Die Standardnamen der Tests lauten `test1x2`, `test1x1` und `test1`. Öffnen Sie die drei Dialogfenster `Sequence Transaction`, und ändern Sie dieses drei Namen in `test1`, `test2` und `test3` zur Verdeutlichung.

1. Wählen Sie `Data` ⇒ `Continuous` ⇒ `Real64 Slider` aus und platzieren Sie dieses Objekt links vom `Sequencer`. Ändern Sie den Namen auf die Eingabeaufforderung `Select Num:`, verkleinern Sie das Objekt und verbinden Sie es mit dem Eingangsanschluss des `Sequencer`.

   Tip: Sie können die Größe eines Objekts beim Positionieren ändern, indem Sie es anklicken und die Ecken des Objekts mit der linken Maustaste ziehen.

2. Wählen Sie eine `AlphaNumeric`-Anzeige aus, platzieren Sie sie unter dem `Sequencer`, verbreitern Sie sie und verbinden Sie sie mit dem Ausgangsanschluss `Log` am `Sequencer`.

3. Speichern Sie das Programm als `seqdat1`. Wählen Sie eine Zahl im `Real64 Slider`-Objekt aus und führen Sie `seqdat1` aus. Es sollte aussehen wie in Abbildung 9-7.

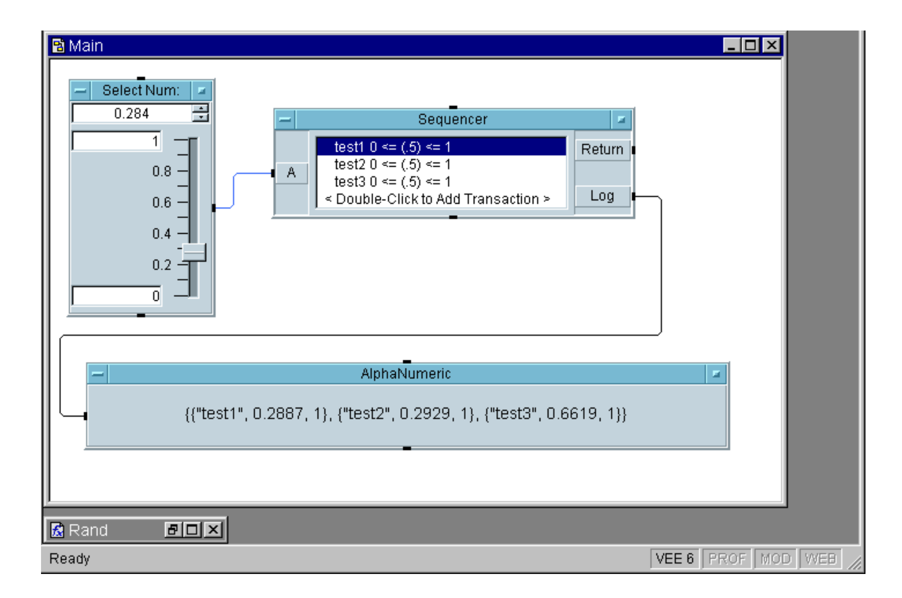


Abbildung 9-7. Weitergeben von Daten mit einem Eingangsanschluss

Je größer die Anzahl der Tests wird, desto mehr Eingangs-Pins sind für die Weitergabe von Daten mit einem Eingangsanschluss erforderlich. Sie können die Anzahl der Eingangs-Pins reduzieren, indem Sie Datensätze an Eingangsanschlüsse weitergeben und einzelne Felder in den Datensätzen für die verschiedenen Tests verwenden. Sie können auch eine separate `UserFunction` verwenden, um gobale Variablen einzurichten, die dann von anderen `UserFunctions` oder einem Ausdrucksfeld im Programm aufgerufen werden können. Die nächste Übung verdeutlicht dies.

Weitergeben von Daten mit einer globalen Variablen

In der Übung wird das Programm `seqdat1` durch Hinzufügen einer globalen Variablen geändert, die den Parameter an die `UserFunction Rand` weitergibt.

1. Löschen Sie das `Real64 Slider`-Objekt `Select Num`. Löschen Sie den Eingangsanschluss `A` am `Sequencer`.

2. Heben Sie die test1-Transaktionsleiste hervor, öffnen Sie das Objektmenü und klicken Sie Insert Trans... an. Wenn das Feld Sequence Transaction angezeigt wird, klicken Sie TEST an, um die Auswahl auf EXEC umzuschalten, und ändern Sie den Namen in Setup.

   Sie verwenden den EXEC-Modus, da die User Function nur eine globale Variable einrichtet und kein Ergebnis ausgibt, das mit einer Spezifikation getestet werden muss.

3. Ändern Sie das Feld FUNCTION in global() und klicken Sie OK an, um das Dialogfenster zu schließen.

   Als nächstes erstellen Sie die UserFunction global().

4. Wählen Sie Device ⇒ UserFunction aus. Ändern Sie den Namen von UserFunction1 in global.

   Wählen Sie Data ⇒ Continuous ⇒ Real64 Slider aus und platzieren Sie dieses Objekt in der UserFunction. Ändern Sie den Namen in Select Num: und verkleinern Sie das Objekt in der vertikalen Richtung.

   Wählen Sie Data ⇒ Variable ⇒ Set Variable und platzieren Sie dieses Objekt rechts von Real64 Slider.

   Ändern Sie den Namen der globalen Variablen von globalA in a. Verbinden Sie den Real64 Slider mit dem Objekt Set Variable.

Fügen Sie zum Anzeigen der Funktion auf dem Bildschirm eine Einblend-Fensteransicht hinzu, in der der Bediener eine Nummer auswählen kann. Fügen Sie eine Schaltfläche Confirm (OK) hinzu, sodass das Fenster so lange in der Anzeige erhalten bleibt, bis der Bediener eine Auswahl getroffen hat. (Sie können diese Aufgaben auch mit einem Dialogfenster Real Input innerhalb der UserFunction global() ausführen.)

5. Wählen Sie Flow ⇒ Confirm(OK) aus und platzieren Sie dieses Objekt über dem Objekt Real64 Slider. Verbinden Sie den Datenausgangs-Pin OK mit dem Sequenzeingangs-Pin von Real64 Slider.

Hinweis

Wenn Sie die Schaltfläche OK unter dem Objekt Set Variable platzieren, bewirkt dies einen logischen Fehler. Dies liegt daran, dass VEE den alten Wert am Slider an das Objekt Set Variable sendet und wartet, bis die Schaltfläche OK angeklickt wird. Alle im Einblendfenster neu eingegebenen Werte werden ignoriert. Wenn OK *über* dem Real64 Slider verbunden wird, wartet VEE mit dem Setzen der globalen Variablen, bis OK gedrückt wurde und verwendet somit den neuen Slider-Wert. Sie können Show Data Flow einschalten, um die Reihenfolge der Ausführung anzuzeigen.

6. Wählen Sie Display ⇒ Note Pad aus und platzieren Sie das Objekt rechts von der Schaltfläche OK. Geben Sie die folgende Benutzer-Eingabeaufforderung im Note Pad ein:

   ```
   Please select a number for this run of tests 1, 2, and
   3.
   ```

7. Wählen Sie das Note Pad, den Real64 Slider und die Schaltfläche OK aus, indem Sie **Strg** drücken und diese Objekte anklicken. Alle diese Objekte werden jetzt mit einem Schatten versehen, um anzuzeigen, dass sie ausgewählt wurden. Klicken Sie Edit ⇒ Add To Panel an. (Denken Sie daran, dass das Menü Edit auch über die rechte Maustaste in einem offenen Bereich der VEE-Anzeige oder in der Detailansicht eines UserObject bzw. einer UserFunction verfügbar ist.) Verkleinern Sie in der Fensteransicht das Fenster, und positionieren Sie das Note Pad oben, den Real64 Slider in der Mitte und die Schaltfläche OK unten.

Hinweis

Wenn Sie die Objekte in der Fensteransicht neu positionieren, hat dies *keine* Auswirkung auf das Layout der Objekte in der Detailansicht.

Öffnen Sie das Fenster UserFunction Properties. Klicken Sie im Ordner General unter Pop-up Panel, um Show Panel on Execute auszuwählen. Abbildung 9-8 zeigt die UserFunction in der Detailansicht, und Abbildung 9-9 zeigt die Fensteransicht.

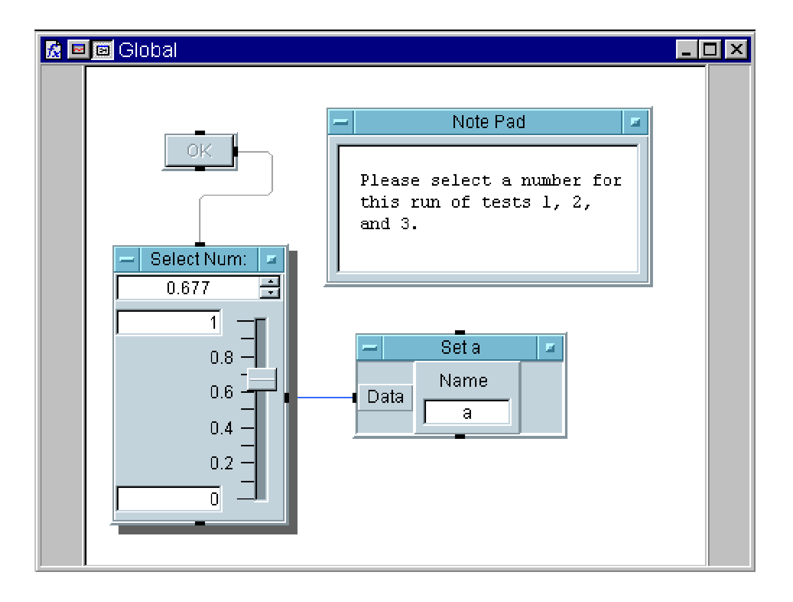


Abbildung 9-8. Die UserFunction "Global" (Detailansicht)

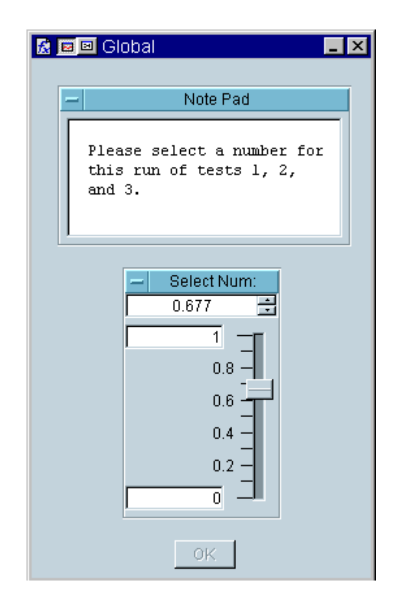


Abbildung 9-9. Die UserFunction "Global" (Fensteransicht)

8. Speichern Sie das Programm als `seqdat2` und führen Sie es aus. Wenn das Einblendmenü angezeigt wird, wählen Sie einen Wert aus und drücken Sie `OK`. Es sollte aussehen wie in Abbildung 9-10.

Hinweis Das Einblendmenü erscheint standardmäßig in der Mitte der Anzeige. Zum Verschieben klicken und ziehen Sie die Titelleiste.

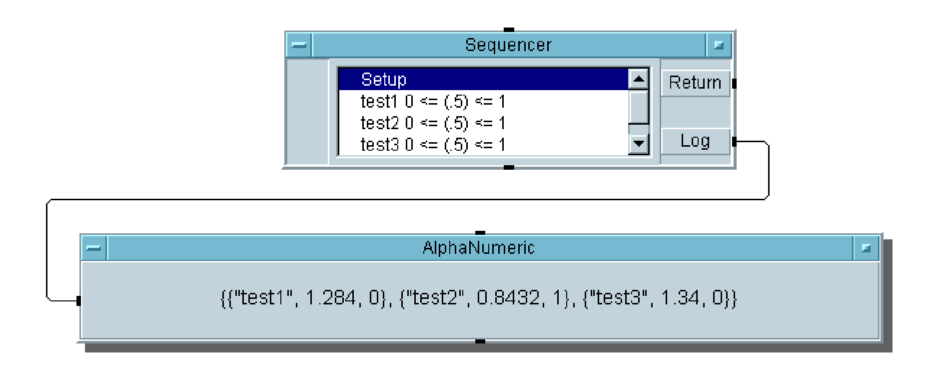


Abbildung 9-10. Weitergeben von Daten über eine globale Variable

Vergleichen einer Wellenformausgabe mit einer Maske

In dieser Übung erstellen Sie eine `UserFunction` mit dem Namen `noisyWv` und rufen sie von einer einzelnen Transaktionszeile im `Sequencer` aus auf. Der Bediener kann die Amplitude der Welle von `0` bis `1` variieren. Diese Funktion simuliert ein Testergebnis, das eine gestörte Wellenform zurückgibt. Sie verwenden das Objekt `Coord` im Menü `Data` ⇒ `Constant` zum Erstellen einer geraden Linienmaske bei `0.6`, die der `Sequencer` zum Testen der gestörten Wellenform verwendet.

1. Erstellen Sie die `UserFunction` mit dem Namen `noisyWv` wie in Abbildung 9-11 in der `Detailansicht` gezeigt.

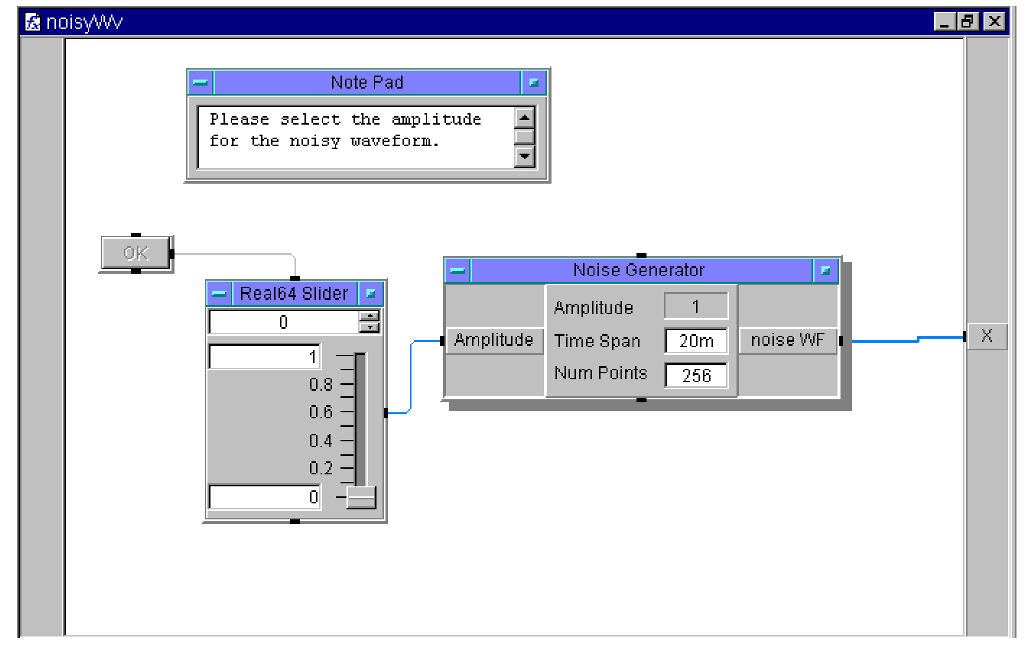


Abbildung 9-11. Die UserFunction "noisyWv" (Detailansicht)

2. Drücken Sie **Strg** und klicken Sie die Schaltfläche `OK`, den `Real64 Slider` und das `Note Pad` an, um diese Objekte zum Erstellen einer Fensteransicht hervorzuheben. Wählen Sie `Edit` ⇒ `Add To Panel` aus.

   Wenn die `Fensteransicht` angezeigt wird, ordnen Sie die Objekte wie gewünscht neu an und passen Sie die Größe des Fensters entsprechend an.

   Öffnen Sie das Objektmenü, klicken Sie `Properties` an und klicken Sie unter `Pop-up Panel` neben `Show Panel on Execute`.

   Die `Fensteransicht` sollte aussehen wie in Abbildung 9-12.

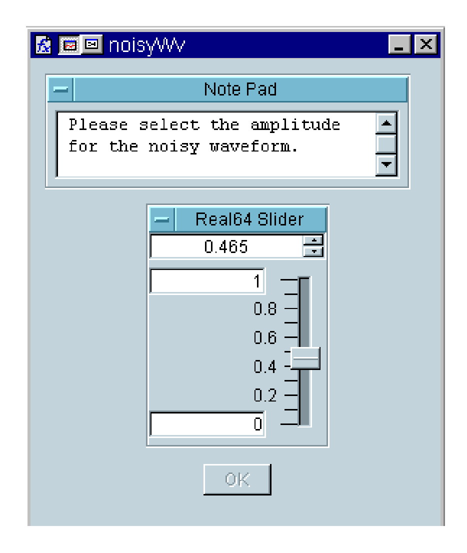


Abbildung 9-12. Das UserObject "noisyWv" (Panel)

3. Wählen Sie `Device` ⇒ `Sequencer` aus und platzieren Sie dieses Objekt links in der Mitte des Hauptfensters. Fügen Sie einen Dateneingangsanschluss hinzu und nennen Sie ihn `mask`.

4. Klicken Sie die Transaktionsleiste an, um das Dialogfenster `Sequence Transaction` aufzurufen. Ändern Sie die Felder wie folgt:

FUNCTION	Geben Sie `noisyWv()` ein.
RANGE:	Klicken Sie und wählen Sie `LIMIT` im Einblendmenü aus. Behalten Sie `<=` bei, und geben Sie `mask` als Namen des Anschlusses `LIMIT` ein. Alle weitere Standardwerte sind in Ordnung; klicken Sie daher `OK` an.

`test1` ruft ein Ergebnis von `noisyWv()` ab und vergleicht es mit dem Grenzwert am Eingangsanschluss `mask`. Wenn die gestörte Welle an allen Stellen kleiner oder gleich der Maske ist, gilt der Test als erfolgreich. Andernfalls gilt er als fehlgeschlagen.

5. Wählen Sie Data ⇒ Constant ⇒ Coord aus und platzieren Sie das Objekt über dem Sequencer. Verbinden Sie seinen Ausgang mit dem Eingangsanschluss mask des Sequencer.

   Öffnen Sie das Coord-Objektmenü, klicken Sie Properties an und stellen Sie die Felder wie folgt ein:

Configuration	Wählen Sie "1D Array" aus.
Size	Geben Sie 2 ein (Sie brauchen nur zwei Koordinatenpaare zur Angabe einer geraden Linie.)

   Klicken Sie OK an.

6. Im Coord- Objekt sehen Sie jetzt zwei Indizes für Koordinatenpaare. Klicken Sie doppelt auf den ersten Index, 0000:, um einen Cursor anzuzeigen. Geben Sie die Koordinaten durch Kommas getrennt ein; VEE fügt die Klammern automatisch hinzu. Geben Sie 0, 0.6 **Tab** 20m, 0.6 ein und klicken Sie anschließend den Arbeitsbereich außerhalb des Objekts ein. Die Einträge lauten wie folgt:

   - Die x-Achse (Zeitachse) für den Noise Generator in noisyWv() reicht von 0 bis 20 Millisekunden; daher die beiden x- Werte 0 und 20m.

   - Die beiden y-Werte sind jeweils 0.6, da Sie eine Maske mit einer geraden Linie erstellen wollen.

Hinweis

Sie können eine beliebige Masken-Wellenform erstellen, indem Sie die entsprechende Anzahl von Koordinatenpaaren konfigurieren und ausfüllen.

Der Vergleichsmechanismus des Sequencer arbeitet genau wie das Objekt Comparator, das den Datentyp Coord zum Testen von Wellenformen akzeptiert. Sie sollten natürlich die beiden Wellenform-Datentypen vergleichen. Drücken Sie **Tab**, um zwischen den Koordinatenpaaren zu wechseln. Wenn Sie fertig sind, klicken Sie den Arbeitsbereich an.

7. Wählen Sie eine AlphaNumeric-Anzeige aus, verbreitern Sie sie, und verbinden Sie sie mit dem Ausgang Log des Sequencer.

8. Speichern Sie das Programm als `seqdat3` und führen Sie es aus. Es sollte aussehen wie in Abbildung 9-13.

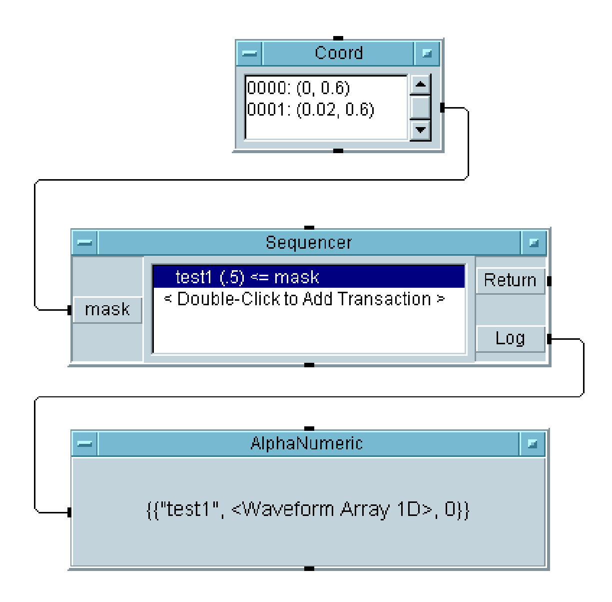


Abbildung 9-13. Vergleichen einer Wellenform mit einer Maske

Die Übungen zum Weitergeben von Daten mit dem `Sequencer` sind damit abgeschlossen. In der nächsten Übung lernen Sie, wie Sie Daten aus verschiedenen Iterationen des `Sequencer` aufrufen und analysieren.

Analysieren von Daten aus dem Sequencer

Wie bereits erwähnt, werden die `Sequencer`-Daten als Datensatz aus Datensätzen ausgegeben. In vielen Fällen jedoch kann der `Sequencer` eine Testserie mehrmals durchlaufen. Dadurch wird ein Array aus Datensätzen generiert. Jeder Datensatz stellt einen Durchlauf des `Sequencer` dar und enthält weitere Datensätze für die einzelnen Tests in einem Durchlauf. Sie können sich dies vorstellen wie einen Quader aus einzelnen kleineren Quadern im Speicher (siehe Abbildung 9-14).

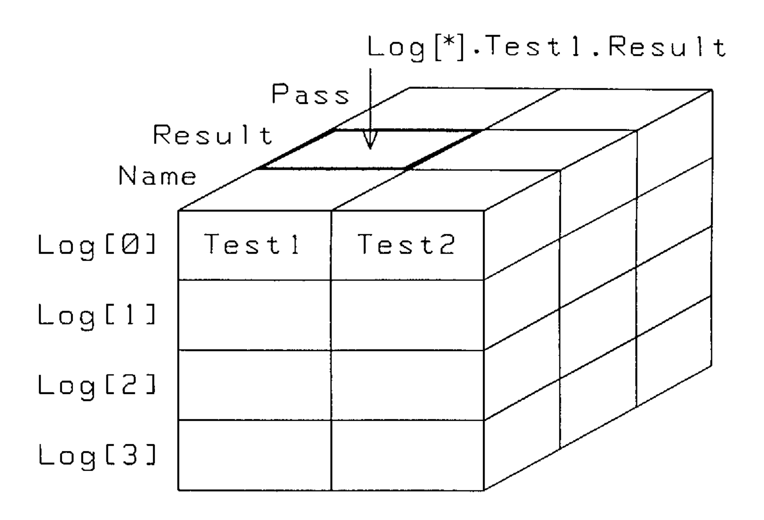


Abbildung 9-14. Ein protokollierter Array aus Datensätzen von Datensätzen

Der Array von Datensätzen wird als `Log` (Protokoll) bezeichnet, da dies der Name ist, der dem Ausgangs-Pin des `Sequencer` zugeordnet wurde. Für den Zugriff auf die Daten einen bestimmten Durchlaufs verwenden Sie die Array-Indexierung mit der Klammernotation.

- `Log[0]` ist der erste Durchlauf am `Sequencer`, `Log[1]` ist der zweite Durchlauf, usw.

- Der Haupt-Datensatz für jeden Durchlauf hat zwei Felder, `Test1` und `Test2`.
- Innerhalb des Datensatzes `Test1` gibt es drei Felder: `Name`, `Result` und `Pass`. Dasselbe gilt für den Datensatz `Test2`.
- `Log.Test1.Result` liefert somit einen Array aus vier Werten; jeder dieser Werte steht für einen der vier Durchläufe. `Log[0].Test1.Result` gibt einen Skalarwert aus, `Result` von `Test1` im ersten Durchlauf (`Log[0]`).

Der protokollierte Array aus Datensätzen vereinfacht das Analysieren und Untersuchen der Daten. Sie könnten sich beispielsweise dafür interessieren, wie viele Tests in einem bestimmten Durchlauf erfolgreich waren. Oder Sie wollen das Durchschnittsergebnis von `Test2` über alle Durchläufe wissen. Oder Sie wollen alle Daten von `Test1` im vierten Durchlauf anzeigen. Alle diese Abfragen sind mit dieser Datenstruktur möglich. In der nächsten Übung werden einige Analyseoperationen mit Daten durchgeführt.

Übung 9-3: Analysieren mehrerer Durchläufe von Daten aus dem Sequencer

1. Löschen Sie den Inhalt der Anzeige und öffnen Sie das Programm `seqdat1.vee`.

Ändern Sie das Programm `seqdat1.vee`, sodass es den `Sequencer` drei Mal durchläuft. Führen Sie anschließend die Analyseoperationen mit den Daten durch.

2. Wählen Sie `Flow` ⇒ `Repeat` ⇒ `For Count` aus und platzieren Sie dieses Objekt über dem Objekt `Real64 Slider`. Ändern Sie die Anzahl der Iterationen in `3` und verbinden Sie den Datenausgangs-Pin mit dem Sequenzeingangs-Pin des Sequencer.

3. Löschen Sie die Datenlinie zwischen dem `Sequencer Log`-Pin und der Anzeige. Wählen Sie `Data` ⇒ `Collector` aus und platzieren Sie dieses Objekt rechts vom `Sequencer`. Verbinden Sie seinen linken oberen Dateneingangs-Pin mit dem `Sequencer Log`-Pin und seinen Pin `XEQ` (links unten) mit dem Sequenzausgangs-Pin am Objekt `For Count`.

Verbinden Sie den Datenausgangs-Pin des `Collector` mit dem Dateneingangs-Pin der `AlphaNumeric`-Anzeige. Vergrößern Sie die Anzeige vertikal, damit genügend Platz für den Array mit drei Elementen zur Verfügung steht.

Der `Sequencer` durchläuft jetzt `test1` und `test2` drei Mal und erfasst die Daten in einem Array mit drei Elementen, von denen jedes einen Datensatz aus Datensätzen für jeden Durchlauf enthält. (Sehen Sie sich hierzu die Quaderdarstellung in Abbildung 9-14 an.)

Führen Sie das Programm an dieser Stelle aus, um die Anzeige der `Sequencer`-Daten zu sehen.

Verwenden Sie jetzt das `Formula`-Objekt zum Extrahieren eines Teils der zu analysierenden Daten. Diese Übung verwendet die Ergebnisse von `test1` für alle drei Durchläufe als Beispiel und findet den Mittelwert aus diesem Array.

4. Wählen Sie `Device ⇒ Formula` aus und platzieren Sie das Objekt unter der Anzeige. Verbinden Sie den Eingangs-Pin von `Formula` mit dem Ausgang des `Collector`. Ändern Sie das Eingabefeld `Formula` wie folgt: `a[*].test1.result`. Verbinden Sie ein Objekt `mean(x)` mit `Formula` und eine `AlphaNumeric`-Anzeige mit `mean(x)`.

   Das "a" bezieht sich auf den Array am Eingangsanschluss `A`. `Test1.result` ruft das richtige Feld auf. Alle Durchläufe werden in einem Array angezeigt. (`A[0].test1.result` bezieht sich beispielsweise nur auf den ersten Durchlauf.)

5. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 9-15.

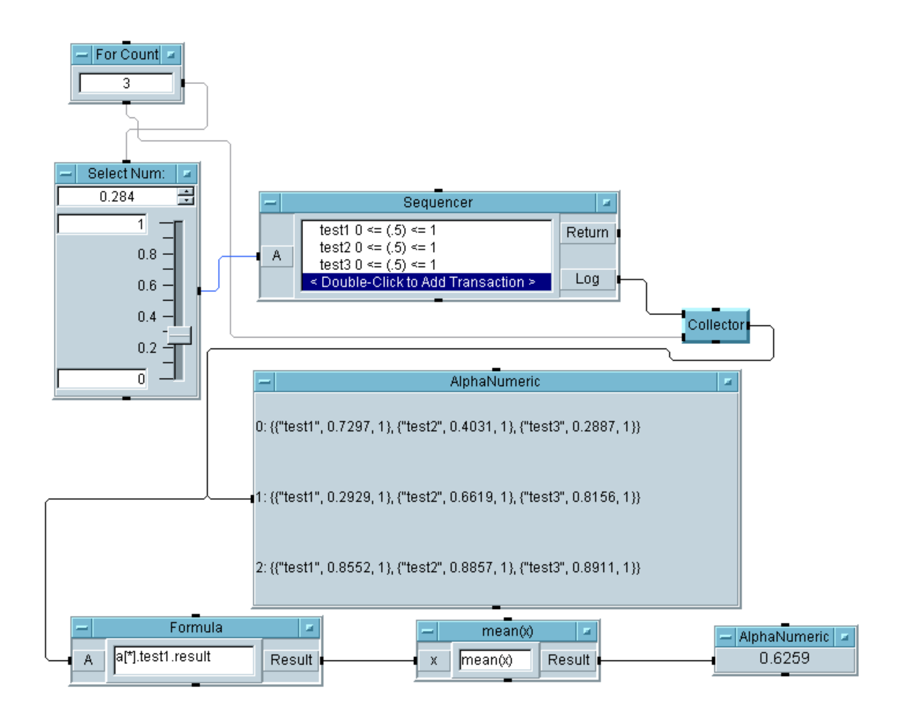


Abbildung 9-15. Analysieren mehrerer Durchläufe von Sequencer-Daten

Diese Übung greift nur auf einen einzelnen Array zu; das Prinzip ist jedoch beim Extrahieren weiterer Daten-Arrays aus dem `Sequencer`-Ausgang das gleiche. Sie können leicht ändern, welche Felder gespeichert werden, indem Sie den `Logging folder` (Protokoll-Ordner) im Dialogfenster `Sequencer Properties` öffnen.

Speichern und Abrufen protokollierter Daten

Diese Übung zeigt die Verwendung der Objekte `To/From File` und `To/From DataSet`.

Übung 9-4: Verwenden der Objekte "To/From File" mit protokollierten Daten

1. Öffnen Sie die Datei `seqdat2` und löschen Sie die Datenlinie zu der Anzeige.

2. Wählen Sie `Flow` ⇒ `Repeat` ⇒ `For Count` und platzieren Sie dieses Objekt links vom `Sequencer`. Ändern Sie die Anzahl in `For Count` in `3` und verbinden Sie seinen Datenausgangs-Pin mit dem Sequenzeingangs-Pin des `Sequencer`.

3. Vergrößern Sie den Arbeitsbereich vertikal und platzieren Sie die `AlphaNumeric`-Anzeige im unteren Bereich. Wählen Sie `Data` ⇒ `Collector` aus und platzieren Sie dieses Objekt im linken Arbeitsbereich. Verbinden Sie den `Sequencer Log`-Pin mit dem Dateneingangs-Pin des `Collector`. Verbinden Sie den Sequenzausgangs-Pin des Objekts `For Count` mit dem `Collector XEQ`-Pin.

   Der `Collector` erstellt einen Array aus Datensätzen von Datensätzen aus dem `Sequencer`. Durch Verwenden der Transaktion `WRITE CONTAINER` im Objekt `To File` können Sie sehr leicht einen beliebigen VEE-Daten-Container in eine Datei schreiben.

4. Wählen Sie `I/O` ⇒ `To` ⇒ `File` aus und platzieren Sie dieses Objekt rechts vom `Collector`. Wählen Sie `I/O` ⇒ `From` ⇒ `File` aus und platzieren Sie dieses Objekt unter dem Objekt `To File`. Fügen Sie dem Objekt `To File` einen Eingangsanschluss hinzu und verbunden Sie den `Collector`-Ausgang damit. Verbinden Sie den Sequenzausgangs-Pin von `To File` mit dem Sequenzeingangs-Pin von `From File`. Verbinden Sie den Datenausgang von `From File` mit der Anzeige.

Markieren Sie `Clear File At PreRun & Open` in `To File` und konfigurieren Sie eine Transaktion `WRITE CONTAINER a`. Konfigurieren Sie eine Transaktion im Objekt `From File` wie die folgende: `READ CONTAINER x`.

Sie können die Standard-Datendatei zum Speichern verwenden.

5. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 9-16.

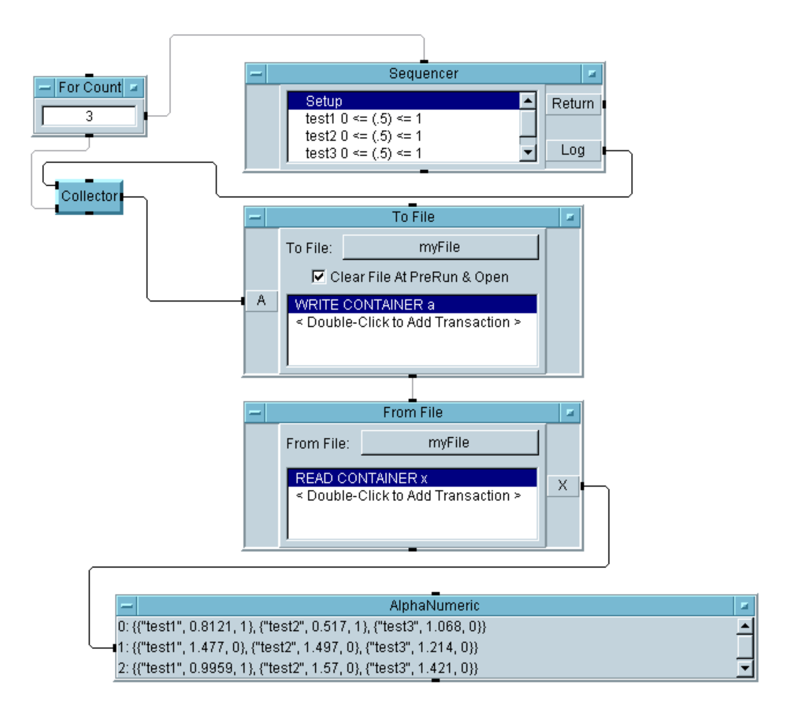


Abbildung 9-16. Speichern von protokollierten Daten mit "To/From File"

Verwenden der Objekte "To/From DataSet" mit protokollierten Daten

Da Sie Testdaten als Datensätze speichern, verwenden Sie eventuell lieber die Objekte `To/From DataSet`. In diesem Fall brauchen Sie keinen `Collector`, da Sie jeden Durchlauf des `Sequencer` an das Ende des `DataSet` anhängen.

Ändern Sie das letzte Programm, sodass es aussieht wie in Abbildung 9-17. Die Objekte `To/From DataSet` befinden sich im Menü `I/O`. Beachten Sie die Sequenzlinie, die zum `From DataSet` führt. Die Sequenzlinie wird verwendet, weil vor dem Auslösen des `From DataSet` gewartet werden soll, bis alle drei Durchläufe an das `DataSet` angehängt wurden.

Ein Grund zur Verwendung der Objekte `To/From DataSet` zum Erfassen von Daten statt der Objekte `To/From File` liegt darin, dass Sie die Daten mit der Funktion `Search Specifier` im Objekt `From DataSet` in nützliche Informationen umwandeln können.

Hinweis Denken Sie daran, das Feld `Get records` in `From DataSet` von `One` in `All` zu ändern.

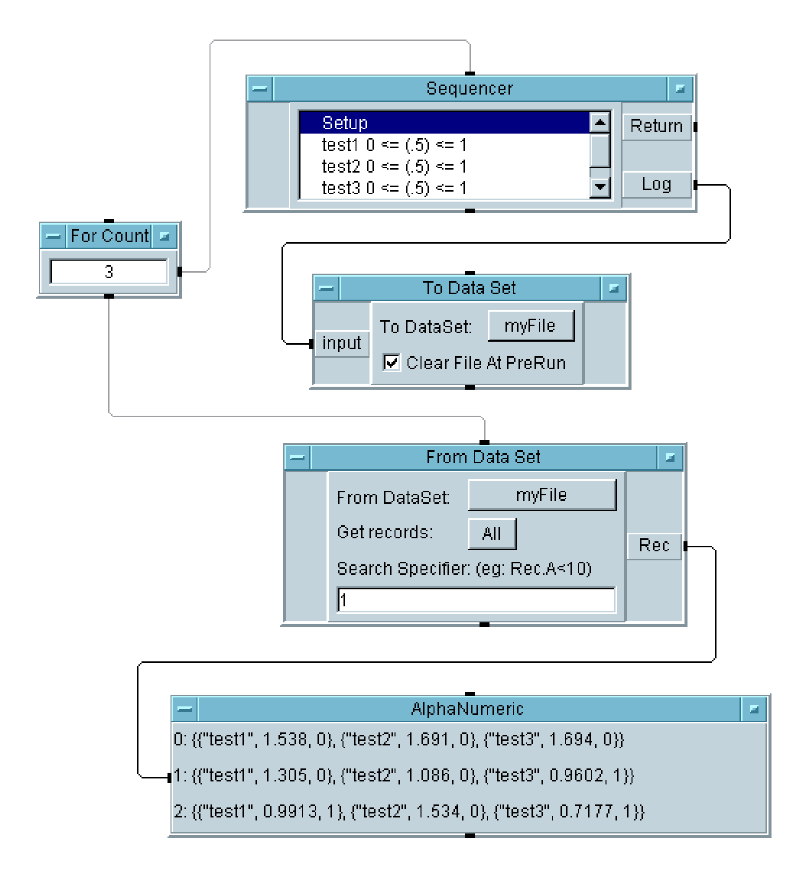


Abbildung 9-17. Speichern von protokollierten Daten mit "To/From DataSet"

Checkliste für das Kapitel

Ermitteln Sie anhand der folgenden Checkliste, ob Sie einzelne Themen noch einmal ansehen sollten, bevor Sie mit dem nächsten Kapitel fortfahren.

- Beschreiben des Konzepts des `Sequencer`-Objekts.
- Konfigurieren eines Tests für den `Sequencer`.
- Hinzufügen, Einfügen und Löschen von Operationen für einen `Sequencer`-Test.
- Aufrufen der vom `Sequencer` protokollierten Daten.
- Verwenden der `Sequencer`-Eingangsanschlüsse zum Weitergeben von Daten an Tests.
- Verwenden von `Globalen Variablen` zum Weitergeben von Daten an Tests.
- Vergleichen einer Wellenformausgabe mit einer Maske.
- Analysieren mehrerer Durchläufe von Daten aus dem `Sequencer`.
- Speichern von Daten mit den Objekten `To/From File`.
- Speichern von Daten mit den Objekten `To/From DataSet`.

10

Verwenden von Bedienerschnittstellen

Verwenden von Bedienerschnittstellen

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Erstellen von Bedienerschnittstellen
- Verwenden von Menüs für einen Bediener
- Importieren von Bitmaps für eine bessere Verständlichkeit
- Sichern von Testprogrammen
- Funktionen von Bedienerschnittstellen
- Verwenden von ActiveX-Steuerelementen zur Erweiterung des Funktionsumfangs von VEE

Erforderliche Zeit für dieses Kapitel: 2 Stunden

Überblick

In diesem Kapitel erfahren Sie mehr über Bedienerschnittstellen wie beispielsweise das Hinzufügen von Menüs, das Anpassen von Schnittstellen, das Hinzufügen von Warnsignalen und das Importieren von Bitmaps. Dieses Kapitel erweitert die Übungen in den vorangegangenen Kapiteln, in denen Sie Bedienerschnittstellen und Einblendfenster erstellt hatten.

Einige Vorteile bei der Verwendung von Funktionen der VEE-Bedienerschnittstelle sind:

- Optimale Bedienerfreundlichkeit
- Verbesserte Programmleistung
- Sicherheit gegen unberechtigte Änderungen
- Gute Übersichtlichkeit durch visuelle Hilfsmittel

Wichtige Punkte zu Bedienerschnittstellen

Dieser Abschnitt bietet einen Überblick zum Erstellen einer Bedienerschnittstelle in VEE.

Erstellen einer Bedienerschnittstelle

VEE umfasst ein breites Spektrum von Auswahl-Steuerelementen, Einblend-Dialogfenstern, Indikatoren und Anzeigen zum Erstellen von Bedienerschnittstellen. Auswahl-Steuerelemente sind beispielsweise Schaltflächen, Auswahlkästchen, Schalter, Dropdown-Menüs und Listenfelder. Indikatoren sind beispielsweise "Tankanzeigen", Thermometer, Füllmarkierungen, VU-Meter und farbige Alarmanzeigen.

Zusätzlich zu den Elementen für Bedienerschnittstellen in VEE können Sie auch Elemente aus anderen Quellen einbeziehen. Es stehen Tausende von Elementen für Bedienerschnittstellen zur Verfügung, die Sie vom World Wide Web herunterladen können. Auch über Active-Steuerelemente stehen Elemente für Bedienerschnittstellen zur Verfügung. (Manche dieser Elemente können kostenlos heruntergeladen werden, für andere muss eine Gebühr entrichtet werden.)

Ob Sie die Objekte für Bedienerschnittstellen aus VEE verwenden oder andere Elemente verwenden, der Prozess zum Erstellen einer Bedienerschnittstelle ist immer der gleiche.

Zum Erstellen einer Bedienerschnittstelle für ein VEE-Programm erstellen Sie eine Fensteransicht des Programms.

1. Wählen Sie das gewünschte Objekt bzw. die Objekte in der Fensteransicht aus, indem Sie die Taste **Strg** gedrückt halten und die einzelnen Objekt zum Auswählen anklicken.

2. Wählen Sie `Edit ⇒ Add To Panel` aus. Die Anzeige wechselt in die Fensteransicht (normalerweise blau angezeigt); diese Ansicht umfasst die in der Detailansicht hervorgehobenen Objekte.

Sie haben jetzt eine Ansicht des VEE-Programms, die Sie so anpassen können, dass Sie nur die Komponenten anzeigt, die der Bediener sehen muss.

Wechseln zwischen der Fensteransicht und der Detailansicht

Zum Wechseln zwischen der Fensteransicht und der Detailansicht eines VEE-Programms klicken Sie das Fenster oder das Detailsymbol in der *Titelleiste* des Fensters an, wie in Abbildung 10-1 gezeigt.

Hinweis Sie müssen eine Fensteransicht des Programms erstellen, damit die Schaltfläche "Fensteransicht" in einer Fenster-Titelleiste angezeigt wird.

Normalerweise entwickeln Sie das Programm in der Detailansicht und erstellen anschließend eine Fensteransicht für die Bedienerschnittstelle. Die Schaltfläche "Fensteransicht" kann in der Titelleiste eines `UserObject`-Fensters, eines `UserFunction`-Fenster oder des Hauptfensters enthalten sein.

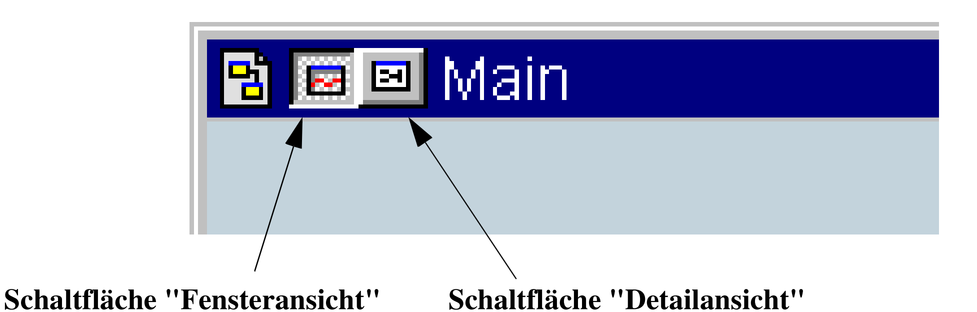


Abbildung 10-1. Schaltflächen "Fenster" und "Detail" in der Symbolleiste

Anpassen einer Bedienerschnittstelle

In der Fensteransicht eines VEE-Programms können Sie die Größe von Objekten ändern, Objekte neu anordnen und die Art der Anzeige von Objekten ändern, ohne *dass sich dies auf die Detailansicht dieser Objekte auswirkt*. Sie können beispielsweise die Titelleiste und die Skalen aus der Fensteransicht einer Anzeige `Waveform (Time)` entfernen, ohne dass sich die `Detail`ansicht dieser `Waveform (Time)` dadurch ändert. Wenn Sie jedoch ein Objekt in der Detailansicht löschen, wird es auch aus der Fensteransicht entfernt.

In der Fensteransicht können Sie verschiedene Farben und Schriften auswählen, um bestimmte Komponenten von anderen abzuheben. Außerdem können Sie zur Verdeutlichung skalierbare Bitmap-Dateien einfügen. Sie können die Fensteransicht auch für den Bediener dokumentieren, indem Sie die Titelleisten über die Objekte `Note Pad` und `Label` bearbeiten, und Sie können in den Objektmenüs die Option `Description` zur Beschreibung der Objekte verwenden.

Abbildung 10-2 zeigt einige der in VEE verfügbaren Indikatoren.

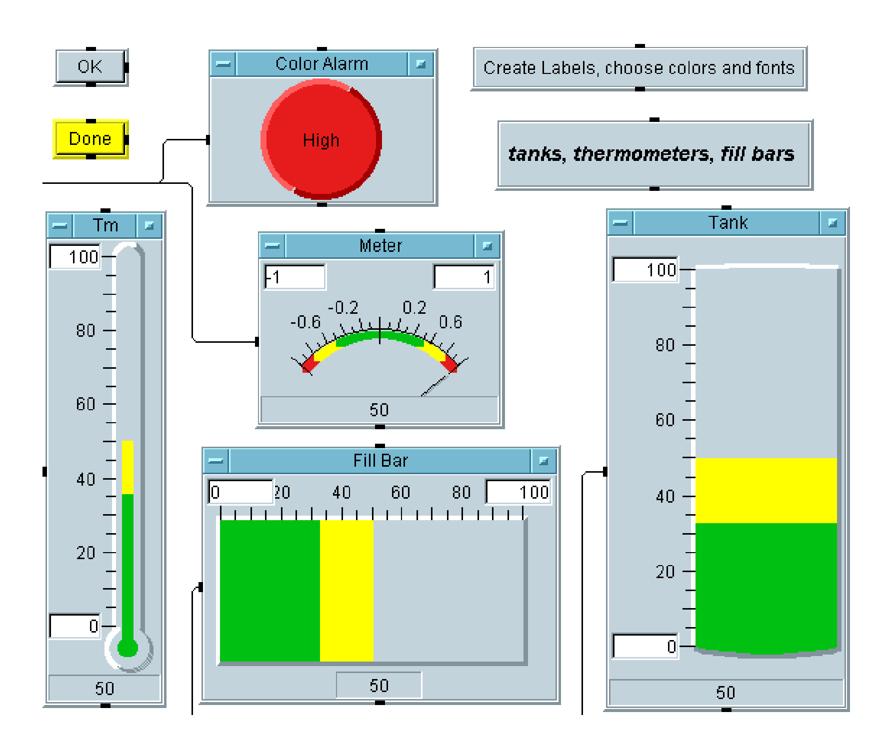


Abbildung 10-2. Eine Auswahl der VEE-Indikatoren

Verwenden der Objekte der Bedienerschnittstelle

In diesem Abschnitt werden die Objekte der Bedienerschnittstelle und die in VEE verfügbaren Optionen vorgestellt. Sie können diesen Abschnitt durchblättern, um sich einen Überblick darüber zu verschaffen, welche Elemente Sie beim Erstellen von Bedienerschnittstellen für Programme verwenden und wie Sie diese Elemente anpassen können. Führen Sie anschließend die Übungen durch, um zu sehen, wie Sie Bedienerschnittstellen für einige typische Aufgaben erstellen können.

Farben, Schriften und Indikatoren

- **Farben und Schriften** Sie können die Farben und Schriften über die Auswahl `File ⇒ Default Preferences` oder die Auswahl `Properties` in den jeweiligen Objektmenüs konfigurieren. Die Auswahl der Farben und Schriften hängt von dem Betriebssystem und den installierten Schriften ab.

- **Farbige Alarmmeldungen** Farbige Alarmobjekte befinden sich im Menü `Display ⇒ Indicator`. Sie können für drei verschiedene Bereiche mit Farben und Textmeldungen als Quadrate oder Kreise konfiguriert werden. Alarme werden häufig zum Simulieren einer LED verwendet oder um Bediener über eine Situation zu informieren, die einen Eingriff erfordert.

- **Tankanzeigen, Thermometer, Füllanzeigen, Messelemente** Diese Objekte befinden sich im Untermenü `Display ⇒ Indicator`. Sie können mit Farben und Beschriftungen angepasst werden. Diese Indikatoren können in horizontalen und vertikalen Formate definiert werden. Sie umfassen drei Bereiche, die über `Properties` in den Objektmenüs konfiguriert werden können.

Grafikbilder

Sie können Bitmaps in die Fensteransicht importieren, indem Sie mit `Background Picture` im Ordner `Panel` des Fensters `Properties` das Hintergrundbild angeben. VEE importiert Dateien in den Formaten `*.jpeg`, `*.png`, `*.wmf`, `*.xwd`, `*.GIF`, `*.bmp` und `*.icn` als Hintergrundbild für das Hauptfenster oder die Fenster `UserObject` und `UserFunction`.

Wenn eine Bitmap-Datei als Hintergrundbild festgelegt wird, erscheinen andere VEE-Objekte auf dem Bild. (Weitere Informationen hierzu finden Sie im Abschnitt “Importieren von Bitmaps für den Fensterhintergrund” auf Seite 409.) Die Bilder können skaliert, nebeneinander angeordnet, zurechtgeschnitten oder zentriert werden. Abbildung 10-3 zeigt ein VEE-Logo als Hintergrundbild mit einer angepassten Größe.



Abbildung 10-3. Logo als Hintergrundbild

Abbildung 10-4 zeigt ein Hintergrundbild mit der Option "nebeneinander anordnen".



Abbildung 10-4. Hintergrundbild, nebeneinander angeordnet

Im Menü `Display` steht außerdem ein Objekt `Picture` zur Verfügung, über das Sie eine Bitmap-Datei in einem Programm platzieren können. Abbildung 10-5 zeigt ein Bild, das mit `Display` ⇒ `Picture` eingebunden und anschließend in VEE zurechtgeschnitten wurde.

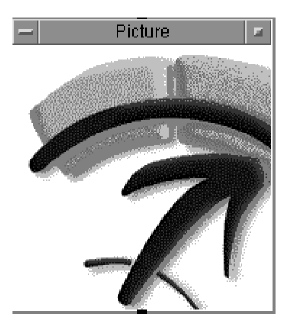


Abbildung 10-5. Ein in VEE zurechtgeschnittenes Bild

Hinweis

Sie können über das Register `Properties` ⇒ `Icon` auch Bitmap-Bilder für beliebige Symbole ändern.

Anzeigen eines Steuerelements für die Bedienereingabe

Es gibt verschiedene Möglichkeiten, ein Programm so einzurichten, dass es über die Eingabe vom Bediener gesteuert werden kann. Sie können die Bedienereingabe über Einblendfenster, beliebige Datenkonstanten, Regler ("Slider") oder Knöpfe ("Knobs") abfragen. Zur Auswahl eines Steuerelements sehen Sie in den Menüs wie `Data` ⇒ `Selection Control`, `Data` ⇒ `Toggle Control` und `Data` ⇒ `Continuous` nach. Abbildung 10-6 zeigt eine Sammlung der Objekte, die Sie zur übersichtlicheren Darstellung für den Bediener anzeigen können.

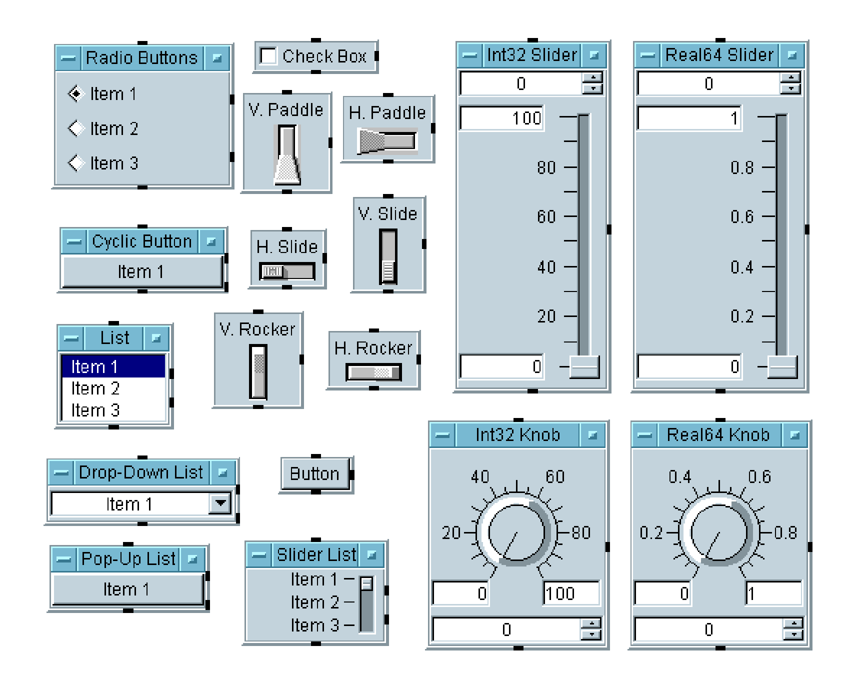


Abbildung 10-6. Steuerelemente aus verschiedenen Data-Untermenüs

Sie können das Aussehen und Verhalten aller in Abbildung 10-6 dargestellten Objekte anpassen. Sehen Sie sich beispielsweise das Dialogfenster `Real64 Knob Properties` in Abbildung 10-7 an. Wählen Sie zum Konfigurieren des Objekts einen Ordner wie beispielsweise `Colors` aus und treffen Sie Ihre Auswahl.

Hinweis

Mit ActiveX können Sie auch Steuerelemente und Anzeigen aus anderen Anwendungen verwenden, wie im Beispiel in “Verwenden eines ActiveX-Steuerelements” auf Seite 417 gezeigt.

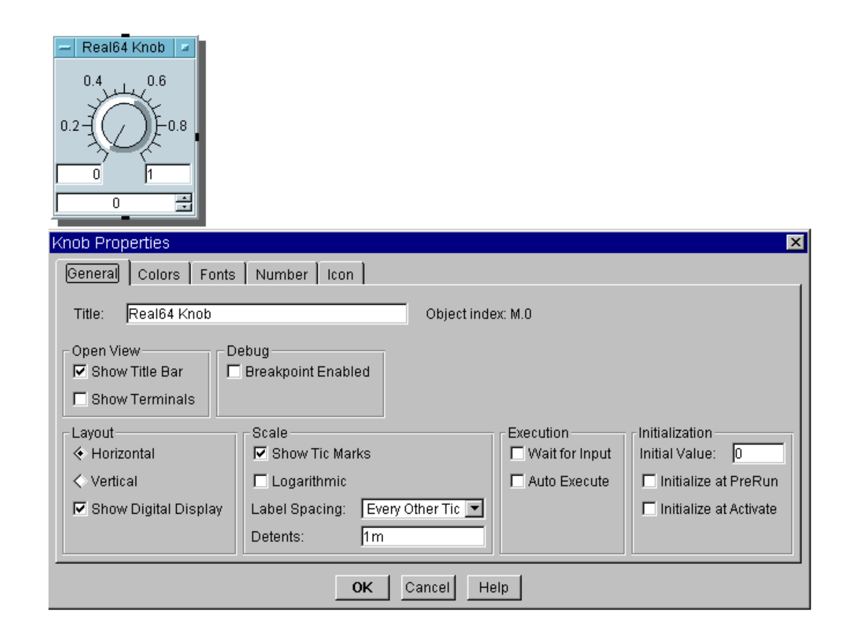


Abbildung 10-7. Das Dialogfenster "Properties"

Anzeigen eines Dialogfensters für die Bedienereingabe

VEE enthält integrierte Einblend-Dialogfenster mit automatischer Fehlerprüfung, Eingabeaufforderungen und Fehlermeldungen. Diese befinden sich unter `Data ⇒Dialog Box`.

Ein Programm kann beispielsweise bei der Ausführung eines Programms vom Bediener die Eingabe einer realen Zahl anfordern. Sie können in das Programm beispielsweise ein Objekt `Real64 Input` einbinden, das bei der Ausführung des Programm automatisch ein Fenster `Real64 Input` für den Bediener anzeigt. Das Fenster `Real64 Input` zeigt außerdem automatisch eine Fehlermeldung an, wenn der Bediener auf die Eingabeaufforderung nicht die richtigen Informationen eingibt. Abbildung 10-8 zeigt das in das

Programm einzubindende Objekt und das bei der Ausführung des Programms angezeigte Fenster `Real64 Input`.

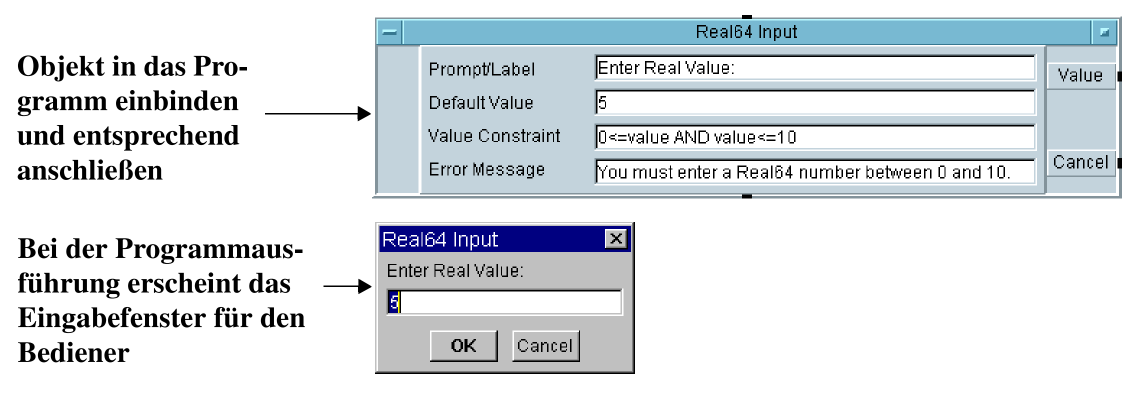


Abbildung 10-8. Ein Texteingabefenster

Abbildung 10-9 zeigt die konfigurierbare Fehlermeldung an, die bei der Programmausführung erscheint, wenn der Bediener `OK` anklickt, ohne die richtigen Informationen im Fenster `Real64 Input` einzugeben.

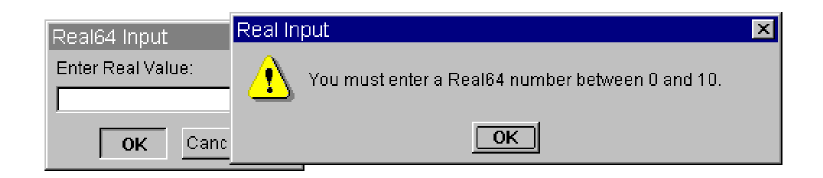


Abbildung 10-9. Ein Beispiel für die automatische Fehlerprüfung

Die Eingabefenster für `Int32` und `Text`, die Sie auch unter `Data` ⇒ `Dialog Box` finden, ähneln dem Eingabefenster `Real64 Input`. Außerdem enthält das Menü `Data` ⇒ `Dialog Box` Auswahlmöglichkeiten für die Auswahl von `Meldungsfenstern`, `Listenfeldern` und `Dateinamen`.

Abbildung 10-10 zeigt ein Dialogfenster, das zur Anzeige einer Meldung eingeblendet wird.

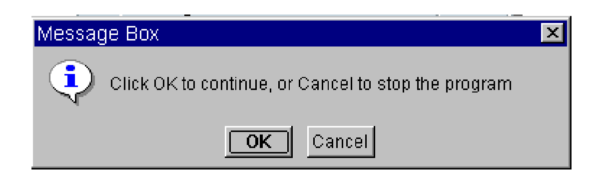


Abbildung 10-10. Das Meldungs-Einblendfenster

Abbildung 10-11 zeigt ein Dialogfenster, das eingeblendet wird, damit der Bediener eine Liste eingeben kann.

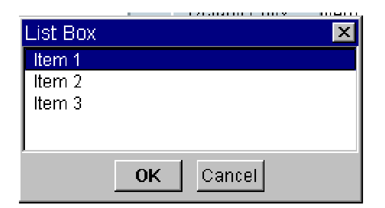


Abbildung 10-11. Das Listenauswahlfenster

Abbildung 10-12 zeigt ein Dialogfenster, das eingeblendet wird, damit der Bediener einen Dateinamen eingeben kann.

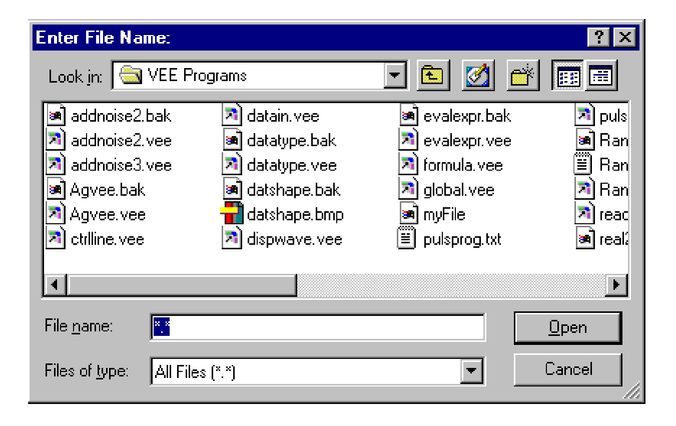


Abbildung 10-12. Ein Einblendfenster zur Dateiauswahl

Anzeigen eines Umschalt-Steuerelements für den Bediener

VEE umfasst integrierte Steuerelemente, die zum Senden von 0 oder 1 verwendet werden können. Zur Verwendung eines Umschalt-Steuerelements legen Sie seinen Anfangsstatus fest und führen ein Unterprogramm aus, wenn das Umschaltelement aktiviert wird. Sie können auch angepasste Bitmaps für `Umschaltelemente` verwenden.

Wenn Sie beispielsweise ein Programm haben, bei dem der Bediener Umschalter oder Alarme festlegen muss, können Sie Umschalt-Steuerelemente verwenden. Abbildung 10-13 zeigt ein Fenster, in dem der Bediener Schalter einstellen kann.

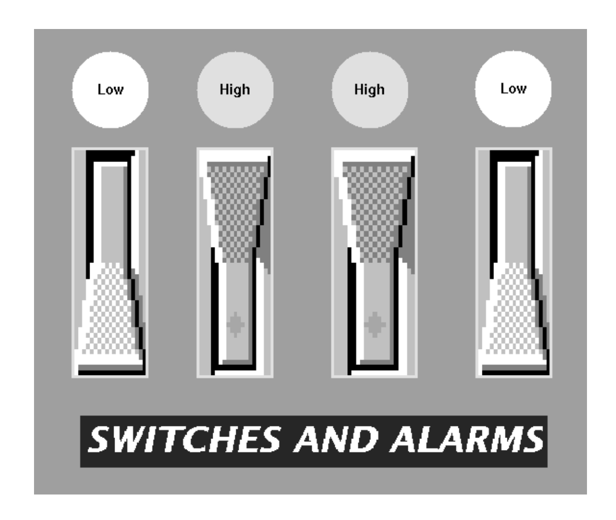


Abbildung 10-13. Kombinierte Umschalter und Alarme

Ausrichten von Objekten in der Bedienerschnittstelle

In der Fensteransicht gibt es eine Funktion "snap-to-grid” (An Gitter anpassen), das die Ausrichtung von Objekten erleichtert. Sie können die Gittergröße von 10 bis 1 ändern (10 ist der Standardwert), um eine genaue Ausrichtung zu ermöglichen (siehe Abbildung 10-14). Mit dieser Funktion geben Sie Ihrem Programm einen professionellen Anstrich. Sie finden die Funktion “Snap-to-grid” im Ordner `Panel` in der Auswahl `Properties`

des Menüs `UserObject` bzw. `UserFunction`. (Denken Sie daran, dass Sie eine Fensteransicht für die Auswahl des Ordners `Panel` erstellt haben müssen, damit das Dialogfenster angezeigt werden kann.)

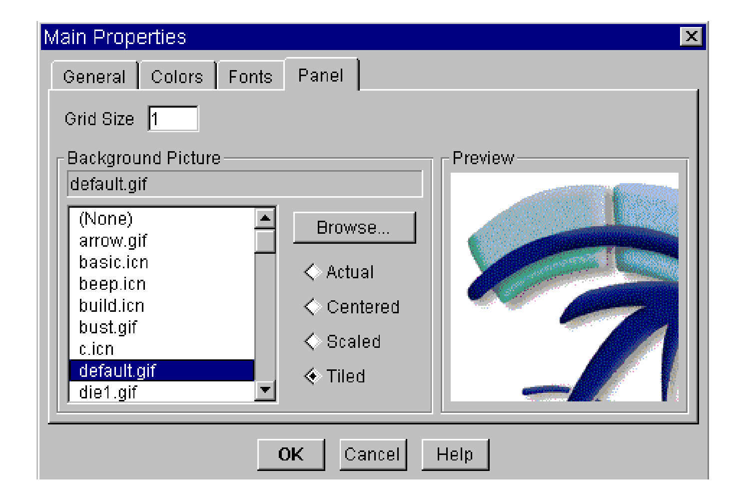


Abbildung 10-14. Konfigurieren der Fenstereigenschaften

Erstellen einer Bedienerschnittstelle für die Tastatur

Sie können mit VEE auch Schnittstellen erstellen, die der Bediener ausschließlich über die Tastatur steuern kann. Diese Schnittstellen erfordern keine Maus.

Sie können beispielsweise das Objekt `OK` als "Softkey" konfigurieren. Normalweise konfigurieren Sie dieses Objekt so, dass es mit einer **F**-Taste verbunden ist. Der Bediener kann dann zur Steuerung des Programms die entsprechenden **F**-Tasten drücken, wie in Abbildung 10-15 gezeigt.



Abbildung 10-15. Ein Softkey zur Ausführung einer UserFunction

Abbildung 10-16 zeigt, wie ein Objekt `OK` zur Verwendung des Dialogfensters `Properties...` konfiguriert wird zur Verbindung mit einer Funktionstaste, mit der **Eingabetaste** oder der Taste **Esc**.

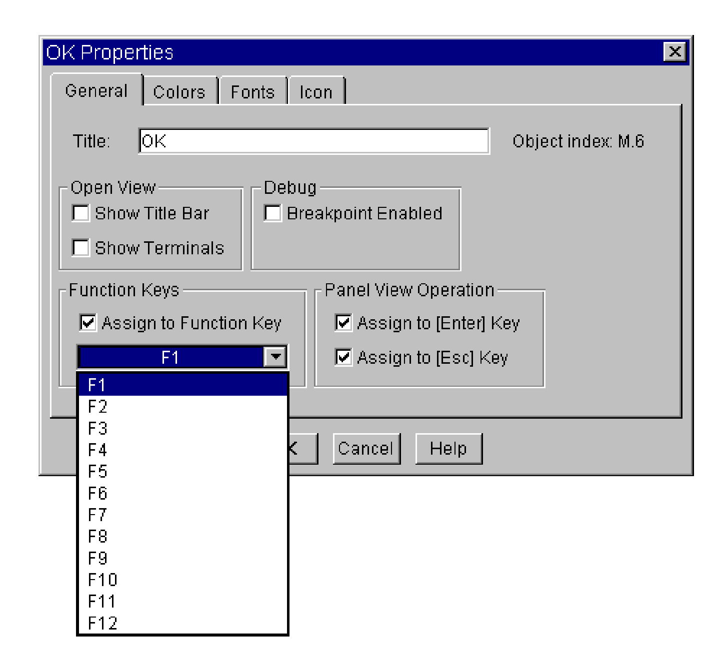


Abbildung 10-16. Konfigurieren des Objekts "Confirm" (OK)

Darüber hinaus kann das Programm über die Tastatur in der Fensteransicht gesteuert werden. VEE hebt automatisch eine Schaltfläche für das Fenster mit einer gepunkteten Linie hervor. Wenn der Bediener die **Eingabetaste** drückt, ist diese Schaltfläche "gedrückt". Ändert der Bediener einen Texteingabebereich, wird mit der **Eingabetaste** diese Änderung übernommen; mit der Taste **Esc** wird die Änderung abgebrochen. Mit der **Tabulatortaste** wird zwischen den verschiedenen Auswahlmöglichkeiten gewechselt und das jeweils aktive Objekt angezeigt. Mit der Kombination **Umschalttaste-Tabulatortaste** wird rückwärts durch diese Optionen gewechselt.

Verwenden Sie die folgenden Kombinationen zum Steuern der Programmausführung:

Strg-G	Ausführen oder fortsetzen
Strg-P	Pause
Strg-T	Schritt

Auswählen der Bildschirmfarben

Verwenden Sie zum Auswählen der Bildschirmfarben das Dialogfenster `File ⇒ Default Preferences`. Stellen Sie die VEE-Umgebung wie gewünscht ein und speichern Sie die Änderungen. Abbildung 10-17 und Abbildung 10-20 zeigen, wie die Bildschirmelemente auf die gewünschte Farbe geändert werden.

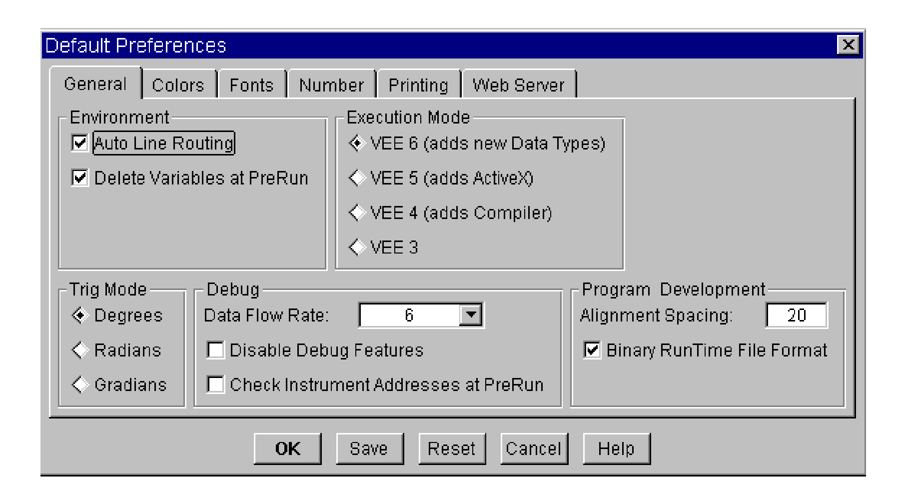


Abbildung 10-17. Das Dialogfenster "Default Preferences"

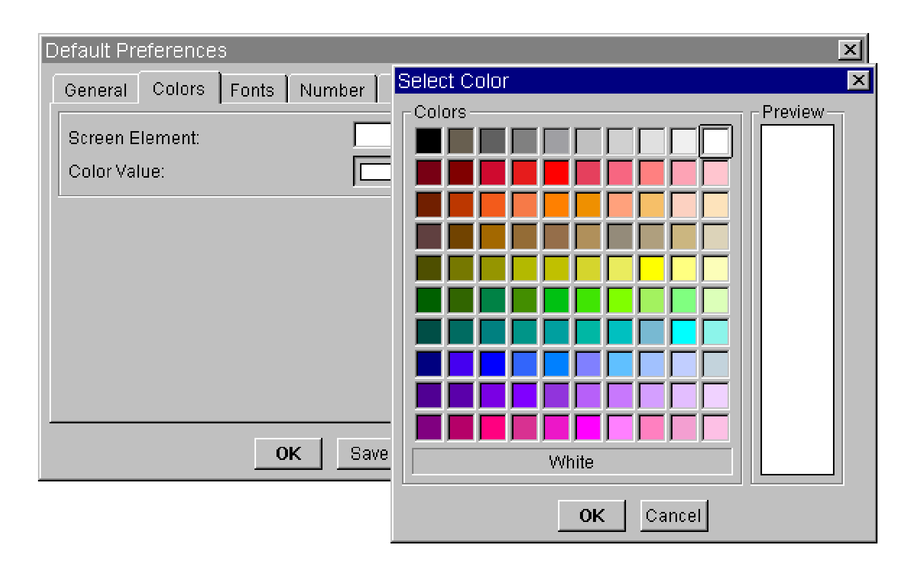


Abbildung 10-18. Farbauswahl für Bildschirmelemente

Sichern eines Programms (Erstellen einer Laufzeit-version)

Damit ein Bediener nicht versehentlich ein Programm löschen kann oder andere Personen sehen können, wie ein Programm arbeitet (durch Anzeige in der Detailansicht), können Sie eine Laufzeitversion eines VEE-Programms erstellen. *Speichern Sie das Originalprogramm und die Laufzeitversion in separaten Dateien.*

Hinweis

Wenn Sie eine Laufzeitversion eines VEE-Programms erstellen, kann die Laufzeitversion *nicht* geändert werden. (Sie können die `Detailansicht` nicht anzeigen.) Erstellen Sie daher Kopien des Originalprogramms, bevor Sie mit diesem Prozess beginnen, und gehen Sie genau anhand der Anweisungen vor.

Führen Sie zum Erstellen einer `Laufzeitversion` eines VEE-Programms die folgenden Schritte aus:

1. Erstellen Sie eine Fensteransicht für die Laufzeitversion des Programms.

2. Speichern Sie das Programm, um eine Kopie zu haben, die Sie ändern können.

3. Wählen Sie `File ⇒ Create RunTime Version...` aus. VEE verwendet automatisch die Erweiterung `*.vxe`, um die Laufzeitversion als solche zu kennzeichnen.

Anzeigen eines Einblendfensters während der Ausführung

Sie können bei der Ausführung eines `UserObject` oder einer `UserFunction` in einem Programm ein Fenster einblenden. Wählen Sie zum Anzeigen eines Einblendfensters `Show Panel on Execute` unter `Properties` im Objektmenü aus. Damit dieses Fenster auf der Anzeige erhalten bleibt, bis der Bediener bereit ist zur Fortsetzung, fügen Sie ein Objekt `Confirm (OK)` hinzu. Andernfalls verschwindet das Fenster, wenn die Ausführung des `UserObject` bzw. der `UserFunction` abgeschlossen ist.

Damit ein Einblendfenster auch über mehrere Aufrufe einer `UserFunction` hinweg angezeigt wird, verwenden Sie die Funktionen `ShowPanel()` und `HidePanel()`. Sie können beispielsweise das angezeigte Einblendfenster als Statusanzeige beibehalten, während das Programm ausgeführt wird. Ein Beispiel hierzu finden Sie im nächsten Abschnitt.

Erstellen einer Statusanzeige

In VEE können Sie Statusanzeigen zur Überwachung der mehrfachen Tests oder Funktionen implementieren. Dies wird mit den Funktionen `ShowPanel()` und `HidePanel()` implementiert, wie in Abbildung 10-19 gezeigt. Weitere Informationen hierzu finden Sie unter “Erstellen einer Statusanzeige” auf Seite 420.

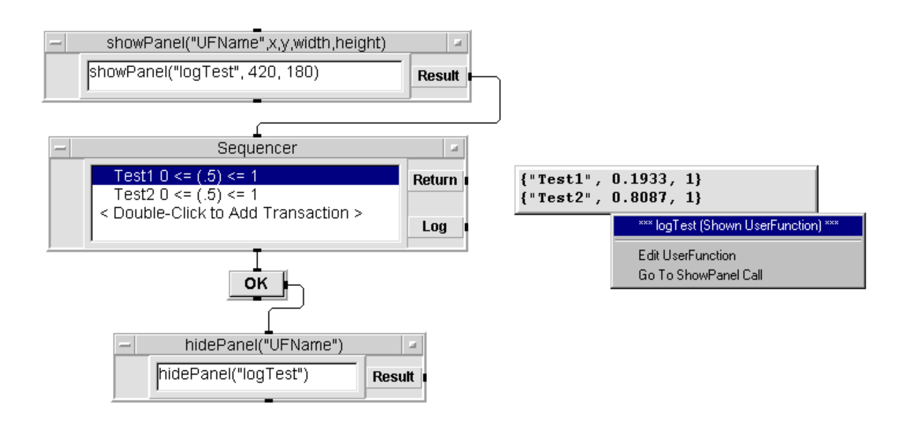


Abbildung 10-19. Erstellen einer Statusanzeige

Allgemeine Aufgaben beim Erstellen von Bedienerschnittstellen

In den folgenden Übungen lernen Sie, wie Sie eine Reihe von Funktionen für die Bedienerschnittstelle implementieren. Insbesondere lernen Sie, wie Sie Menüs, Warnungen und Statusanzeigen erstellen und Bitmaps importieren, um Ihren Programmen optisch mehr Eindruck zu verleihen. In allen diesen Übungen haben Sie die Möglichkeit, die Schnittstellen anzupassen.

Übung 10-1: Verwenden von Menüs

In dieser Übung erstellen Sie eine Bedienerschnittstelle, die ein Menü mit drei Auswahloptionen enthält: `die1`, `die2` und `die3`. Wenn der Bediener eine Auswahl trifft, wird eine Funktion mit dem gleichen Namen aufgerufen, die einen Würfel mit einem, zwei oder drei Punkten anzeigt. Dieses Programm simuliert eine Situation, in der der Bediener einen Test aus einem Menü auswählen muss. Sie lernen außerdem, wie Sie ein Bitmap zum Ändern des Erscheinungsbilds eines Symbols importieren. Das Programm erhält den Namen "Dice" (Würfel).

Beginnen Sie mit dem Erstellen der drei `UserFunctions`.

1. Wählen Sie `Device` ⇒ `UserFunction` aus.

Sie könnten jedes beliebige Symbol zum Anzeigen des importierten Bitmap verwenden; in diesem Beispiel verwenden wir das Objekt `Picture`.

2. Wählen Sie `Display` ⇒ `Picture` aus und platzieren Sie dieses Objekt in der `UserFunction`.

3. Öffnen Sie das `Picture`-Objektmenü, klicken Sie `Properties` an und nehmen Sie die Auswahl `Show Title Bar` unter `Open View` zurück. Wählen Sie `die1.gif` unter `Picture` aus, klicken Sie `Scaled` und anschließend `OK` an.

Hinweis

Klicken Sie zum Aufrufen eines Objektmenüs, wenn `Show Title Bar` ausgeschaltet ist, mit der rechten Maustaste auf das Objekt.

Hinweis Obwohl VEE standardmäßig das Bitmaps-Unterverzeichnis verwendet, können Sie auch Bitmaps aus anderen Verzeichnissen aufrufen.

Sie sollten jetzt ein Bild eines Würfels mit einem Punkt auf der Oberseite haben.

4. Wählen Sie `Flow ⇒ Confirm (OK)` aus und platzieren Sie es unter dem Würfel. Verbinden Sie den Sequenzausgangs-Pin von `Picture` mit dem Sequenzeingangs-Pin von `OK`.

   Wählen Sie die Objekte `Picture` und `OK` aus (drücken Sie **Strg** und klicken Sie die Objekte an, um einen Schatten zu erstellen). Öffnen Sie das Einblendmenü `Edit`, indem Sie den Mauszeiger auf den Hintergrund positionieren und die rechte Maustaste drücken. Wählen Sie `Add to Panel` aus.

5. Ändern Sie die `UserFunction Title` und `Panel Title` in `die1`. Ordnen Sie die Objekte an und ändern Sie ihre Größe nach Bedarf.

Hinweis Zum Verschieben von Objekten in `Panel View` klicken Sie mit der rechten Maustaste das Objekt an und wählen Sie `Move` aus.

   Wählen Sie `Show Panel on Execute` im Dialogfenster `Properties`. Klicken Sie den Ordner `Panel` an und ändern Sie die Gittergröße in `2` für eine präzisere Ausrichtung. Klicken Sie anschließend `OK` an.

6. Erstellen Sie zwei weitere `UserFunctions`, indem Sie `Clone` im `die1`-Objektmenü auswählen. Die neuen `UserFunctions` erscheinen automatisch als `die2` und `die3`. Ändern Sie die Picture-Objekte in `die2.gif` bzw. `die3.gif`. Überprüfen Sie alle Einstellungen der neuen Funktionen, um sicherzustellen, dass sie mit denen von `die1` übereinstimmen (mit Ausnahme der Namen und Bitmaps). Das Programm sollte jetzt aussehen wie in Abbildung 10-20. Stellen Sie die Funktionsfenster als Symbole dar.

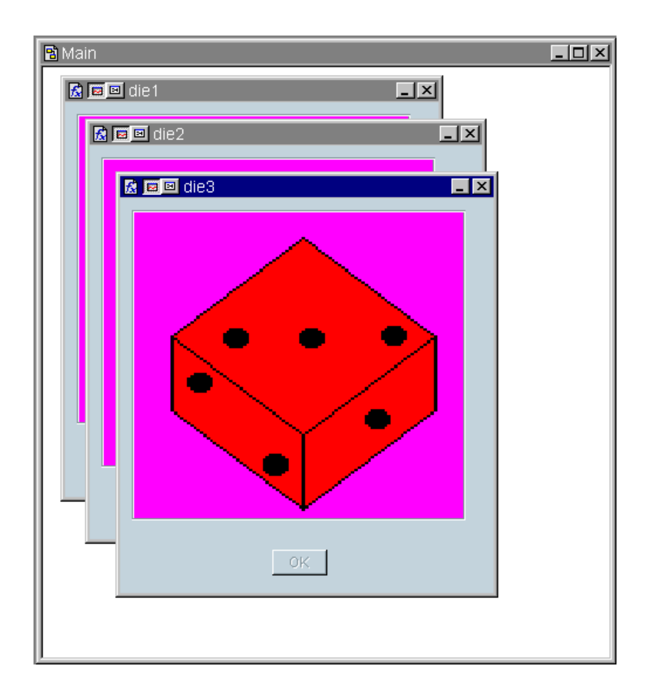


Abbildung 10-20. Eine frühe Phase im Dice-Programm

Erstellen Sie ein Menü, über das eine dieser drei Funktionen zum Aufrufen ausgewählt werden kann.

7. Wählen Sie `Data ⇒ Selection Control ⇒ Radio Buttons` aus.

   `Radio Buttons` ist ein Objekt, das einen Numerierungswert (Datentyp `Enum`; eine Textzeichenfolge mit einer zugeordneten Ordnungszahl) aus der benutzerdefinierten Liste an seinen oberen Ausgangsanschluss ausgibt. Wenn Sie die Liste beispielsweise als Montag, Dienstag, Mittwoch, Donnerstag und Freitag definieren, kann der Bediener den Tag aus einem Menü auswählen; mit `Radio Buttons` würde dann der entsprechende Tag ausgegeben.

   Das erste Element in der Liste ist der Ordnungsposition `0` zugeordnet; das *n*te Elemente der Ordnungsposition *n*`-1`. "Montag" hat somit in der obigen Liste die Ordnungsposition `0` und "Freitag" hat die Ordnungsposition `4`. Die Ordnungspositionen erscheinen am unteren Ausgabe-

anschluss. Eine ausführlichere Erläuterung finden Sie im Help-Eintrag im Objektmenü.

8. Öffnen Sie das Radio Buttons-Objektmenü und wählen Sie Edit Enum Values... aus.

   Geben Sie die Namen der Funktionen die1, die2 und die3 ein, indem Sie mit der **Tabulatortaste** zwischen den einzelnen Felder wechseln, *nicht* jedoch nach dem letzten letzten Feld. Klicken Sie OK an.

Hinweis

Es gibt sechs Menüformate zur Steuerung der Datenauswahl. Radio Buttons zeigt die Einträge als Schaltflächen an. Die Auswahl des Bedieners wird im Textformat als Datentyp Enum ausgegeben. Cyclic Button blättert nacheinander durch die Numerierungswerte, wenn der Bediener die Schaltfläche anklickt. List zeigt alle Numerierungswerte in einer Liste an, wobei das ausgewählte Element hervorgehoben ist. Drop-down list, Pop-up list (Einblendliste) und Slider list (Reglerliste) sind die weiteren drei Optionen.

9. Öffnen Sie das Radio Buttons-Objektmenü, klicken Sie Properties an und wählen Sie Auto Execute aus. Ändern Sie den Titel auf die Eingabeaufforderung: Make a Selection:

Richten Sie ein Call-Objekt ein, sodass der Wert, den der Bediener im Objekt Radio Buttons auswählt, jetzt zu dem Namen der Funktion wird, den das Objekt Call Function aufruft.

10. Klicken Sie Device ⇒ Call an. Wählen Sie Add Terminal ⇒ Control Input und anschließend Function Name aus und klicken Sie OK an. Der Steuer-Pin Function Name akzeptiert einen Wert des Datentyps Enum oder Text als Eingabe. Verbinden Sie den Datenausgangs-Pin Radio Buttons mit dem Eingangsanschluss Function Name des Objekts Call Function. Verbinden Sie den Sequenzausgangsanschluss von Radio Buttons mit dem Sequenzeingangsanschluss von Call Function. Klicken Sie die2 unter Make a Selection: an und beachten Sie, dass sich der Call Function Name in die2 ändert, wie in Abbildung 10-21 dargestellt.

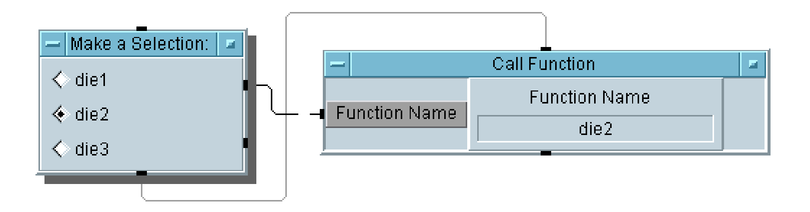


Abbildung 10-21. Das Dice-Programm (Detailansicht)

Hinweis

Der Eingangsanschluss `Call` erfordert ein `Text Scalar`; VEE wandelt daher das `Enum Scalar` in ein `Text Scalar` um.

Denken Sie daran, dass die gepunktete Linie einen Steuer-Pin kennzeichnet. Wenn `Auto Execute` eingeschaltet ist, wird `Radio Buttons` immer ausgeführt, wenn Sie eine Änderung daran vornehmen, und sendet die Auswahl an `Call`. Der Steuer-Pin an `Call Function` ersetzt den Funktionsnamen, sobald der Pin Daten empfängt. Das Objekt `Call` ruft die angegebene Funktion *nicht* auf, bis sein Sequenzeingangsanschluss ausgelöst wird.

Hinweis

Wenn ein Programm `Auto Execute` und seine Sequenzanschlüsse verwendet, braucht der Bediener nicht die Schaltfläche **Ausführen** anzuklicken, um das Programm zu starten.

Fügen Sie eine Bedienerschnittstelle hinzu, die nur die Eingabeaufforderung, das Menü und die Einblendfenster mit den Auswahloptionen anzeigt.

11. Wählen Sie das Objekt `Radio Buttons` aus, indem Sie **Strg** anklicken, und klicken Sie das Zielobjekt an. Wählen Sie anschließend `Edit` ⇒ `Add To Panel` aus.

12. Öffnen Sie das Objektmenü, wählen Sie `Properties` aus und passen Sie die Farben und Schriften wie gewünscht an.

13. Führen Sie das Programm aus, indem Sie eine Auswahl treffen. (Verwenden Sie nicht die Schaltfläche **Ausführen**, da hierdurch die bereits in dem Menü getroffene Auswahl verwendet wird.)

    Das Programm sollte bei der Ausführung aussehen wie in Abbildung 10-22 dargestellt.

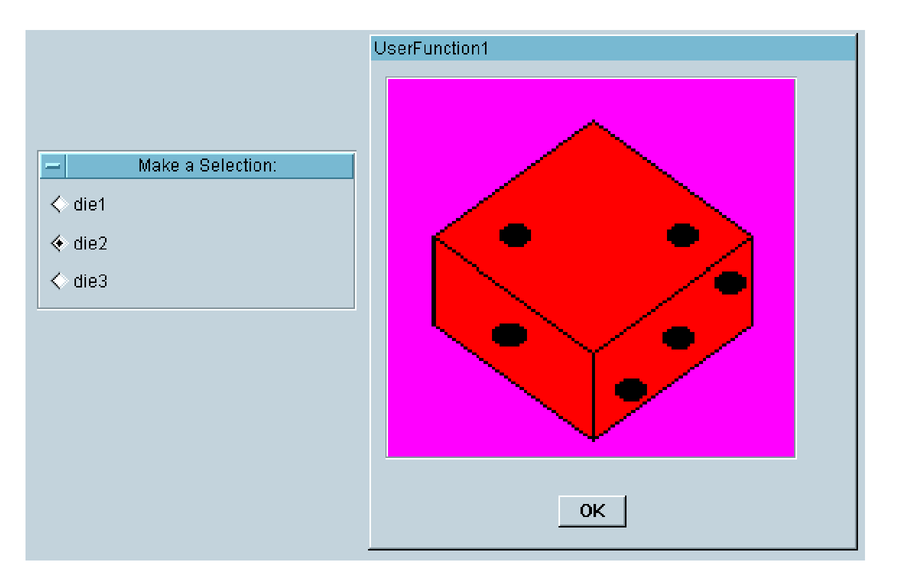


Abbildung 10-22. Das Dice-Programm (Fensteransicht)

Vor der nächsten Übung sollten Sie einige Dinge beachten:

- Sie können die in dieser Übung beschriebenen Techniken beim Erstellen von Menüs für jedes beliebige Programm verwenden.

- `Radio Buttons` können auch zur Auswahl eines Programms in einer kompilierten Sprache verwendet werden; verwenden Sie hierzu das Objekt `Execute Program` mit einem Steuer-Pin (“Command”), der das aufzurufende Programm kennzeichnet. Wenn Sie eine Bibliothek kompilierter Funktionen importiert hatten, können Sie auch das Objekt `Call` zum Ausführen einer Funktion aus der Bibliothek verwenden.

- Sie können dieses Programm auch mit dem Dateneingangsanschluss `File Name` am Objekt `Picture` in einer einzelnen `UserFunction` optimieren und anschließend die entsprechende Bitmap-Datei an das Objekt senden. Wenn Sie viele verschiedene Bitmaps verwenden, ist diese Vorgehensweise bei der Programmierung effizienter.

- Normalerweise verwenden Sie in komplizierteren Programmen **Ausführen** statt `AutoExecute`. Sie können das Programm so gestalten, dass es bei einer Datenkonstante oder einem Auswahl-Steuerobjekt angehalten wird; verwenden Sie hierzu `Wait for Input` statt `AutoExecute`. Weitere Informationen hierzu finden Sie unter `Help`.

Übung 10-2: Importieren von Bitmaps für den Fensterhintergrund

Bitmaps sind für den Inhalt Ihrer Programme nicht von Bedeutung; Sie können jedoch damit Ihre Tests übersichtlicher und professioneller gestalten. Sie können beispielsweise eine Schemazeichnung importieren, die den Inhalt des Tests darstellt. In dieser Übung importieren Sie Bitmaps für Fensterhintergründe, auf denen die Standard-VEE-Objekte platziert werden.

Bitmaps können importiert werden für Symbole, für das Objekt `Picture` oder für Fensteransichts-Hintergründe in `UserObjects` oder `UserFunctions`. Sie erstellen eine Einblend-`UserFunction` mit dem Namen `Bitmap`, die ein Objekt `Label` und ein Objekt `Confirm (OK)` enthält.

1. Wählen Sie `Device ⇒ UserFunction` aus.

2. Wählen Sie `Flow ⇒ Confirm (OK)` und `Display ⇒ Label` aus und platzieren Sie diese Objekte im Fenster `UserFunction`.

3. Ändern Sie den Namen der `UserFunction` in `Bitmap`.

4. Wählen Sie die Objekte `OK` und `Label` aus, um sie mit einem Schatten hervorzuheben. Öffnen Sie das Menü `Edit`, indem Sie den Zeiger auf den Arbeitsbereich der UserFunction positionieren und mit der rechten Maustaste klicken. Wählen Sie `Add to Panel` aus.

5. Öffnen Sie das UserFunction-Menü, wählen Sie `Properties` und anschließend `Show Panel on Execute` aus. (Sie können die Titelleiste anklicken, um das Feld `Properties` aufzurufen.) Heben Sie die Auswahl `Show Title Bar` unter `Pop-up Panel` auf.

   Öffnen Sie den Ordner `Panel`, ändern Sie die `Grid Size` in `2`, wählen Sie `default.gif` und `Scaled` unter `Background Picture` aus und klicken Sie `OK` an.

6. Öffnen Sie das Fenster `Properties` für das Objekt `Label`, und stellen Sie die Werte wie folgt ein:

General/Title:	Ändern Sie den Eintrag in `Bitmap Function`.
Label Justification	Ändern Sie den Eintrag in `Center Justify`.
Colors/Object/ Background	Wählen Sie `Light Gray` aus und `klicken Sie OK an`.
Fonts/Object/Text:	Wählen Sie eine größere Schrift mit fetten Typen aus und klicken Sie `OK` an. Markieren Sie `Automatically Resize Object on Font Change`.
Appearance/Border	Klicken Sie `Raised` an. Klicken Sie `OK` an, um die Änderungen vorzunehmen, und schließen Sie das Dialogfenster `Properties`.

7. Positionieren Sie den Titel `Bitmap Function` und die Schaltfläche `OK` wie gewünscht. Stellen Sie die UserFunction `Bitmap` als Symbol dar.

8. Wechseln Sie zum Hauptfenster. Klicken Sie `Device` ⇒ `Call` und anschließend `Select Function` im Objektmenü an, und wählen Sie `Bitmap` aus. Führen Sie das Programm aus. Das Einblendfenster sollte aussehen wie in Abbildung 10-23.



Abbildung 10-23. Die Bitmap-Funktion

Übung 10-3: Erstellen einer auffälligen Warnung

Diese Übung enthält mehrere verschachtelte `UserFunctions`. Die erste `UserFunction` ist der Alarm selbst, der ein rotes Quadrat anzeigt und einen Signalton ausgibt. Die zweite `UserFunction` ruft den Alarm wiederholt auf und generiert somit einen Blinkeffekt und einen pulsierenden Signalton, bis der Bediener den Alarm ausschaltet.

Beginnen Sie mit der Programmierung der Alarmfunktion.

1. Wählen Sie `Device ⇒ UserFunction` aus. Ändern Sie den Namen in `alarm`.

2. Wählen Sie `Display ⇒ Beep` aus und platzieren Sie dieses Objekt links oberhalb von der `UserFunction`. Passen Sie die Einstellungen so an, dass ein lauter Signalton mit einer Dauer von 1 Sekunde generiert wird. Ändern Sie den Wert im Feld `Duration (sec)` auf `1`. Ändern Sie den Wert im Feld `Volume (0-100)` auf `100`.

Hinweis

Bei diesen Anleitungen wird davon ausgegangen, dass Ihr Computer mit einer Hardware ausgestattet ist, die den Signalton unterstützt. Bei manchen Windows 95, Windows 98, Windows 2000 und Windows NT 4.0-Systemen wurde die `Standard-Systemkonfiguration` für den `Standard-Systemton` geändert.

Hinweis

Sie brauchen das Objekt `Beep` nicht an einer anderen Komponente anzuschließen. Es wird bei der Ausführung der Funktion aktiviert.

3. Klicken Sie `Display ⇒ Indicator ⇒ Color Alarm` an und platzieren Sie dieses Objekt in der `UserFunction`. Öffnen Sie das `Color Alarm` -Objektmenü, klicken Sie `Properties` an und stellen Sie die Felder wie folgt ein: Heben Sie unter `Open View` die Auswahl `Show Title Bar` auf. Klicken Sie unter `Layout` die Option `Rectangular` an. Löschen Sie ggf. unter `Limits/High Test` den Text neben `High Text`. Klicken Sie `OK` an.

4. Klicken Sie `Data ⇒ Constant ⇒ Real64` an, ändern Sie die Angabe in `1` und schließen Sie das Objekt am Eingangsanschluss von `Color Alarm` an. (Dadurch wird der `Alarm` immer auf seinen hohen Bereich eingestellt mit der Standardfarbe rot.)

   Verwenden Sie zur Synchronisierung der Anzeige mit dem Beep-Objekt im Sekundentakt ein Objekt `Delay` (Verzögerung) mit der Einstellung `1` Sekunde.

5. Wählen Sie `Flow ⇒ Delay` aus, stellen Sie den Wert auf `1` ein und schließen Sie seinen Sequenzeingangs-Pin am Sequenzausgangs-Pin von `Color Alarm` an. Der Alarm hält dann eine Sekunde lang an.

6. Wählen Sie `Display ⇒ Note Pad` aus und fügen Sie die folgende Meldung hinzu: `TURN OFF INSTRUMENTS!`. Ändern Sie die Größe des `Note Pad` wie gewünscht.

7. Wechseln Sie zum `Hauptfenster`. Klicken Sie `Device ⇒ Call` an, wählen Sie `Select Function` im `Call`-Objektmenü und anschließend `alarm` aus. Führen Sie das Programm testweise aus. Die Detailansicht des `UserFunction-Alarms` sollte aussehen wie in Abbildung 10-24.

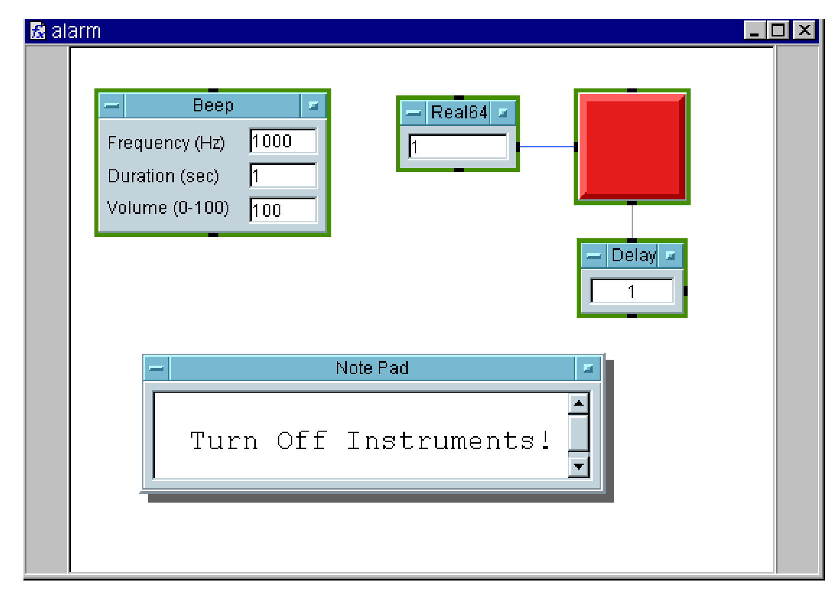


Abbildung 10-24. Die UserFunction Alarm (Detailansicht)

8. Kehren Sie zurück zum Alarmfenster. Wählen Sie die Anzeige `Color Alarm` aus und zeigen Sie das `Note Pad` an. Öffnen Sie das Einblendmenü `Edit` und wählen Sie `Add To Panel` aus. Ordnen Sie in der Fensteransicht die Objekte an und ändern Sie ihre Größe. Öffnen Sie das `Note Pad`-Objektmenü und klicken Sie `Properties` an. Legen Sie die Einträge wie folgt fest:

Open View/ Show Title Bar	Heben Sie diese Auswahl auf.
Editing/Enabled	Heben Sie diese Auswahl auf.
Fonts/Text size	Vergrößern Sie den Text und wählen Sie `Font Style: Bold` (Fettschrift) aus.
Fonts	Wählen Sie `Automatically Resize Object on Font Change` aus.
Appearance/Border	Ändern Sie die Angabe in `Raised border`.

Klicken Sie `OK` an, um das Dialogfenster `Properties` zu schließen.

9. Ändern Sie den `Color Alarm` ebenfalls in die Angabe `Raised Border`.

10. Klicken Sie doppelt auf die Titelleiste der UserFunction, um das Dialogfenster `Properties` aufzurufen, und wählen Sie `Show Panel on Execute` aus. Heben Sie die Auswahl von `Show Title Bar` auf. Ändern Sie die Angabe `Panel Title` in `alarm`. Stellen Sie `Alarm` als Symbol dar.

11. Wechseln Sie zum Hauptfenster und löschen Sie das Objekt `Call`. (VEE hält die Alarmfunktion weiterhin im Speicher. Wenn Sie die Funktion erneut ändern wollen, wählen Sie `Edit` ⇒ `Edit UserFunction` aus oder klicken Sie doppelt auf das Symbol.)

Erstellen Sie die Funktion, die die Alarmfunktion wiederholt aufruft.

12. Wählen Sie `Device` ⇒ `UserFunction` aus und ändern Sie den Namen der `UserFunction` in `warning`.

13. Wählen Sie `Flow` ⇒ `Repeat` ⇒ `Until Break` aus.

14. Wählen Sie `Device` ⇒ `Call` aus, ändern Sie den `Function Name` in `Alarm`, und verbinden Sie ihren Sequenzeingangs-Pin mit dem Datenausgangs-Pin von `Until Break`.

Fügen Sie ein `Check Box`-Objekt hinzu, über das der Bediener gefragt wird, ob der Alarm ausgeschaltet werden soll.

15. Wählen Sie `Data` ⇒ `Toggle Control` ⇒ `Check Box` aus. Öffnen Sie das Fenster `Check Box Properties`, ändern Sie den Namen in `Turn off alarm?`, wählen Sie `Scaled` unter `Layout` aus, wählen Sie `Initialize at PreRun` und vergewissern Sie sich, dass der Wert `0` lautet, vergrößern Sie die Schrift für den Namen und klicken Sie `OK` an. Verbinden Sie den Sequenzausgangsanschluss von `Call` mit dem Sequenzeingangsanschluss von `Check Box`.

    Dadurch wird ein Eingangsobjekt erstellt, das ein `Check Box` verwendet. Wenn der Bediener dieses Auswahlkästchen anklickt, erscheint ein `X`, und das Objekt gibt `1` aus; andernfalls gibt das Objekt `0` aus. Die Ausgabe kann mit einem `If/Then/Else`-Objekt getestet werden, um VEE anzuweisen, was als nächstes ausgeführt werden soll.

16. Wählen Sie `Flow ⇒ If/Then/Else` aus und platzieren Sie dieses Objekt rechts von `Toggle`. Verbinden Sie den Datenausgang von `Toggle` mit dem Dateneingang `A` des Objekts `If/Then/Else`. Ändern Sie den Ausdruck im `If/Then/Else`-Objekt in: `a == 1`. (Das Symbol für *"ist gleich"* lautet `==`, nicht `=`.) Wenn der Anschluss `A` eine `1` enthält, löst der `Then`-Ausgang aus; andernfalls löst der `Else`-Ausgang aus.

    Verbinden Sie den Ausgang von `Toggle` mit dem Eingang von `If/Then/Else`.

17. Wählen Sie `Flow ⇒ Repeat ⇒ Break` aus und verbinden Sie dieses Objekt mit dem `Then`-Ausgang am Objekt `If/Then/Else`, wie in Abbildung 10-25 gezeigt.

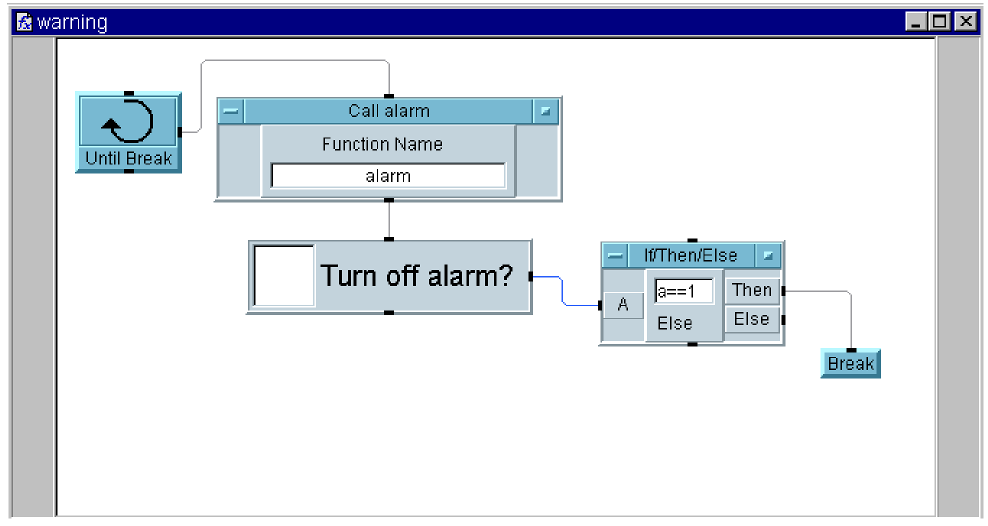


Abbildung 10-25. Die UserFunction "Warning" (Detailansicht)

18. Wählen Sie das Objekt `Check Box (Turn off alarm?) aus`, indem Sie die rechte Seite des Objekts anklicken. Öffnen Sie das Einblendmenü `Edit` und wählen Sie `Add To Panel` aus. Ändern Sie die Größe der Fensteransicht, sodass sie die `Check Box` umschließt.

19. Öffnen Sie das Fenster `Warning UserFunction Properties`, wählen Sie `Show Panel on Execute` aus und heben Sie die Auswahl von `Show Title` auf (dieser Titel erfüllt für den Bediener keinen Zweck). Klicken Sie `OK` an.

20. Wechseln Sie zum Hauptfenster und klicken Sie `Device` ⇒ `Call` an, öffnen Sie sein Objektmenü, klicken Sie `Select Function` und anschließend `Warning` an. Verschieben Sie das `Call`-Objekt nach oben in die Mitte der Anzeige. Stellen Sie das Hauptfenster als Symbol dar.

21. Standardmäßig zeigt VEE die Alarm- und Warnungsfenster in der Mitte des Bildschirms an, sodass der Alarm über dem Auswahlkästchen zum Stoppen des Alarms blinkt. Da beide Bildschirmpositionen nicht gesperrt sind, können Sie sie auf dem Bildschirm anders anordnen, indem Sie die Einblendfenster an die gewünschte Position ziehen. Mit dem blinkenden Alarmfenster ist dies allerdings etwas schwierig. Klicken und ziehen Sie statt dessen die Kante des Fensters. Falls erforderlich, stoppen Sie das Programm über die Schaltfläche "Stopp" in der Symbolleiste. Führen Sie das Programm aus.

Wenn Sie die beiden Fenster wie in Abbildung 10-26 gezeigt positioniert haben, können Sie das Programm durch Anklicken des Felds neben der Eingabeaufforderung `Turn off alarm?` stoppen.



Abbildung 10-26. Das Programm "Warning"

Übung 10-4: Verwenden eines ActiveX-Steuerelements

Diese Übung zeigt die Verwendung eines ActiveX-Steuerelements in VEE. Sie können ActiveX-Steuerelemente aus anderen Anwendungen in VEE-Programme einbinden. In diesem Fall binden Sie ein Steuerelement `ProgressBar` (Fortschrittsleiste) ein und verwenden eine Schleife, um den Verlauf in dieser Leiste von 0% bis 100% anzuzeigen. Dieses allgemeine Prinzip gilt auch für andere ActiveX-Steuerelemente.

1. Klicken Sie `Device` ⇒ `ActiveX Control References...` an und wählen Sie `Microsoft Windows Common Controls 6.0` aus. Klicken Sie `OK` an.

2. Klicken Sie `Device` ⇒ `ActiveX Controls` ⇒ `ProgressBar` an. Vergrößern Sie das Objekt `ProgressBar`. Öffnen Sie sein Objektmenü und beachten Sie, dass der Name `ProgressBar` lautet. VEE hat automatisch eine deklarierte Variable erstellt, die sich auf das ActiveX-Steuerelement beziehen. Sie können den Namen `ProgressBar` in `Formula`-Ausdrücken verwenden, genau wie mit anderen Variablen oder Dateneingängen.

3. Klicken Sie `Device ⇒ Formula & Object Browser` an, wählen Sie `ActiveX Objects`, `Library: MSComcltLib`, `Class: ProgressBar`, `Members: Value` aus und klicken Sie `Create Set Formula` an. Platzieren Sie das Objekt oben in der Mitte des `Haupt-fensters`.

4. Für die Schleife von 1 bis Hundert und zur Anzeige des Verlaufs fügen Sie ein Objekt `For Range` hinzu. Wählen Sie `Flow ⇒ Repeat ⇒ For Range` aus, platzieren Sie das Objekt unter dem `ProgressBar` und stellen Sie die Werte wie folgt ein: `From: 0`, `Thru: 100` und `Step:10`. Verbinden Sie den Ausgang von `For Range` mit dem Eingangsanschluss `Value` von `ProgressBar`.

5. Sie können die Ausführung des Programms verlangsamen, um die Aktualisierung der `ProgressBar` besser sehen zu können. Wählen Sie hierzu `Flow ⇒ Delay` aus und platzieren Sie das Objekt rechts vom Objekt `For Range`. Wählen Sie den Wert `.2` aus. Verbinden Sie den Sequenzausgangs-Pin der `ProgressBar` mit dem Sequenzeingangs-Pin des Objekts `Delay` (siehe Abbildung 10-27) und führen Sie das Programm aus.

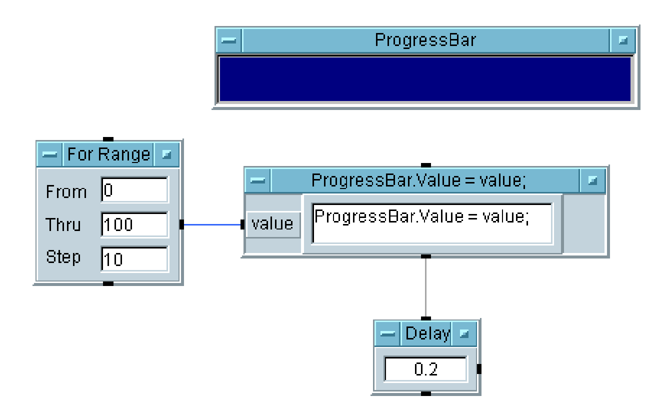


Abbildung 10-27. Verwenden des ActiveX-Steuerelements “ProgressBar”

Hinweis

ActiveX-Steuerelement `Objektmenüs` haben sowohl `Properties` als auch `Control Properties`. Das Menü `Properties` legt die VEE-Eigenschaften des Objekts fest. Die `Steuerelement-Eigenschaften` werden über das Steuerelement bereitgestellt; sie können sich je nach Typ des ActiveX-Steuerelements unterscheiden.

Untersuchen Sie alle mit VEE ausgelieferten Beispiele für Steuerelemente, um die Arbeitsweise dieser Elemente besser zu verstehen. Sie können auch mit weiteren Steuerelementen und Anzeigen anderer Hersteller die Funktionen der Benutzerschnittstelle in VEE erweitern.

Abbildung 10-28 zeigt ein weiteres Beispiel von VEE mit einem Steuerelement aus MS Chart. Nach der Auswahl einer Steuerelement-Bibliothek über das Dialogfenster `Device` ⇒ `ActiveX Controls References` können Sie mit dem `Function & Object Browser` oder dem Objekt `Declare Variable` die Eigenschaften und Methoden eines Steuerelements verwenden.

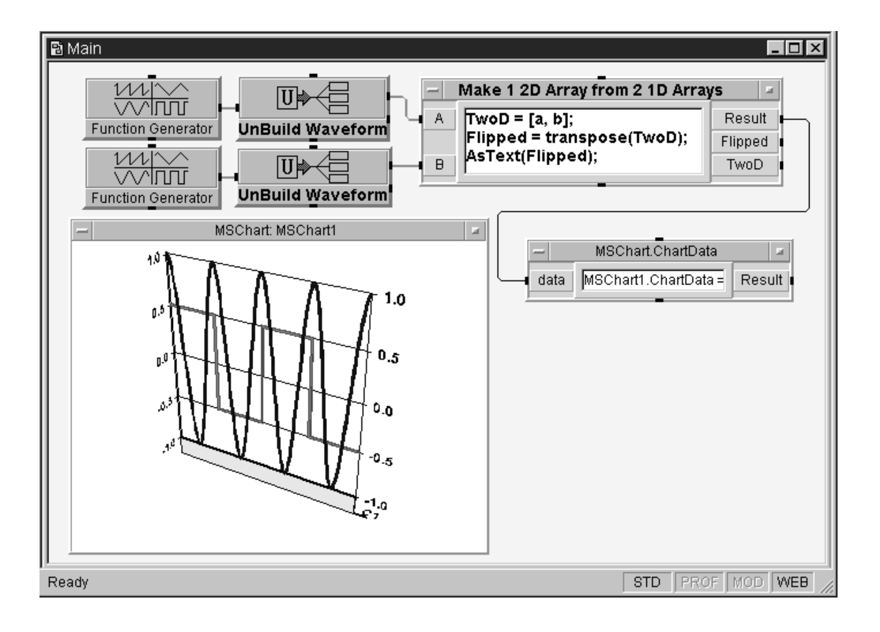


Abbildung 10-28. Beispiel eines ActiveX-Steuerelements mit MSChart

Übung 10-5: Erstellen einer Statusanzeige

In dieser Übung lernen Sie, wie Sie die Funktionen aus dem "Dice"-Programm zum Erstellen einer Statusanzeige verwenden können. (Sie finden die Übung zum "Dice"-Programm in “Verwenden von Menüs” auf Seite 403.) Normalerweise wird dieses Programm mit dem `Sequencer`-Objekt verwendet, wenn eine Reihe von Tests ausgeführt werden sollen und Sie die zurückgegebenen Ergebnisse anzeigen wollen. Sie verwenden die Funktion `random()`, die bei Verwendung der Standardeinstellungen einen Real-Wert zwischen `0` und `1` zurückgibt.

1. Klicken Sie `Device` ⇒ `Sequencer` an. Klicken Sie doppelt auf die Transaktionsleiste, und konfigurieren Sie Ihren Test mit dem Standardnamen `test1` und ersetzen Sie das Feld `FUNCTION:` durch `random()`. Siehe Abbildung 10-29.

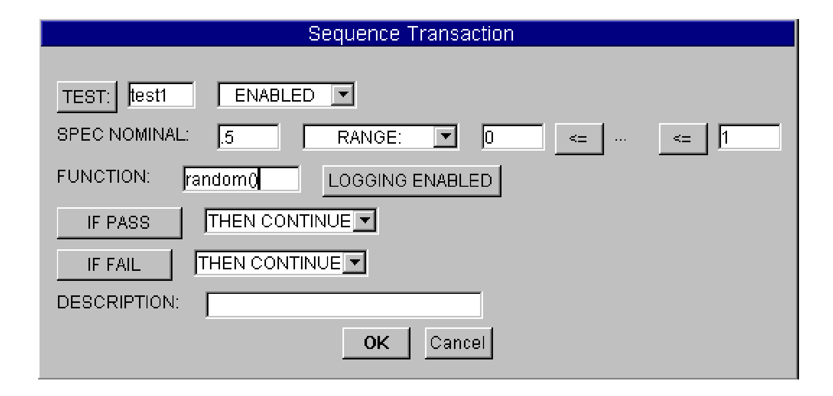


Abbildung 10-29. Konfigurieren von Test1

2. Konfigurieren Sie einen zweiten Test `test2` auf die gleiche Weise.

3. Öffnen Sie das Fenster `Sequencer Properties`, wählen Sie das Register `Logging` aus. Wählen Sie unter `Logging Mode` die Option `Log Each Transaction To: logTest(thisTest)` aus und klicken Sie `OK` an.

   Bei der Ausführung der einzelnen Transaktionen erstellt der `Sequencer` für jeden Test einen Datensatz mit dem Namen "`thisTest`"; die Felder

dieses Datensatzes können über das gleiche Register konfiguriert werden. Sie können eine UserFunction mit dem Namen `"logTest"` (oder einem anderen Namen) erstellen, damit der `Sequencer` die UserFunction `LogTest()` mit dem Datensatz `thisTest` am Ende jeder Transaktion ausführt. Auf diese Weise können Sie die Statusanzeige aktualisieren.

4. Wählen Sie `Device` ⇒ `Function & Object Browser` ⇒ `Built-in Functions` ⇒ `Panel` ⇒ `showPanel` aus und platzieren Sie dieses Objekt über dem `Sequencer`. Löschen Sie die Eingangs-Pins und ändern Sie die Parameter in `"logTest",420,180`, wobei Sie die beiden letzten Parameter weglassen. Verbinden Sie den Ausgangs-Pin `Result` von `showPanel` mit dem Sequenzeingangs-Pin des `Sequencer`. `ShowPanel` gibt bei einer erfolgreichen Ausführung eine `1` aus.

   `LogTest` ist der Name der `UserFunction`. Die beiden anderen Parameter sind die `X`- und `Y`-Koordinaten auf dem Bildschirm, ausgehend von der linken oberen Ecke. Diese Angabe weist VEE an, wo das `User Function`-Fenster bei der Anzeige platziert werden soll. (Die Fensterabmessungen sind in diesem Beispiel nicht enthalten.)

5. Erstellen Sie die `UserFunction` mit dem Namen `logTest` wie nachfolgend gezeigt. Fügen Sie einen Eingangs-Pin hinzu. (Der Protokoll-`Datensatz` wird als Eingang verwendet.) Legen Sie das Objekt `Logging AlphaNumeric` auf dem Fenster ab und verbinden Sie es mit dem Eingangs-Pin.Wählen Sie `Logging AlphaNumeric` aus und klicken Sie `Edit` ⇒ `Add to Panel` an. Passen Sie in der Fensteransicht die Größe und Platzierung wie gewünscht an. Heben Sie die Auswahl der Anzeige und der Fenster-Titelleisten auf.

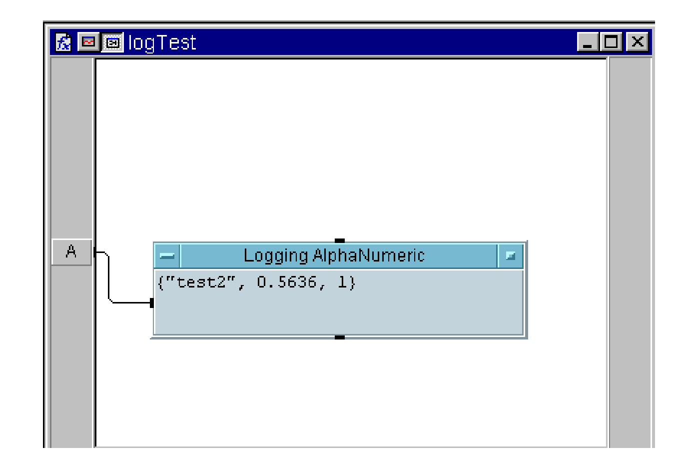


Abbildung 10-30. Die UserFunction "LogTest" (Detail)

Abbildung 10-31 zeigt die Fensteransicht nach der Ausführung des Programms. Diese Darstellung gibt Ihnen eine Vorstellung von der Anzeige des Programms.

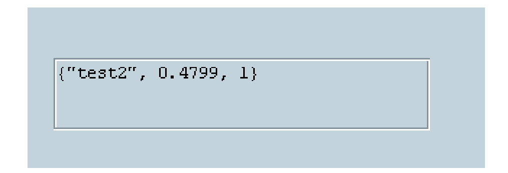


Abbildung 10-31. Die UserFunction "LogTest" (Fensteransicht)

6. Wählen Sie `hidePanel` im Fenster `Function & Object Browser` und anschließend `Flow` ⇒ `Confirm(OK)` aus. Ändern Sie den Parameter `hidePanel()` in `logTest`. Löschen Sie den Eingangs-Pin. Verbinden Sie die Objekte wie in Abbildung 10-32 gezeigt.

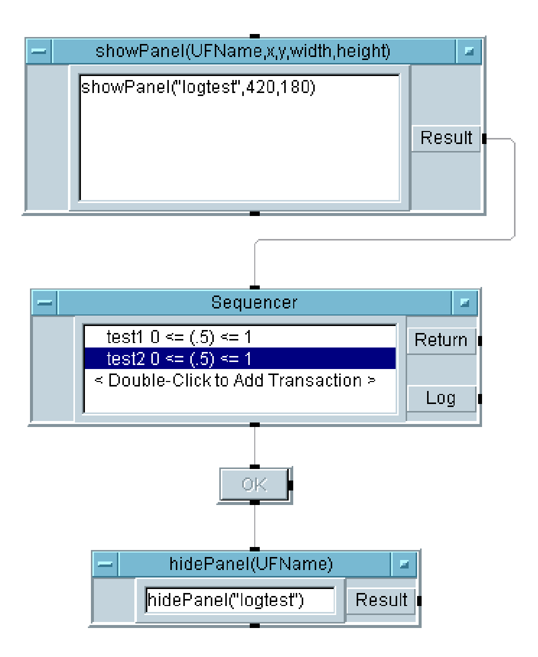


Abbildung 10-32. Statusanzeigeprogramm (vor der Ausführung)

7. Führen Sie das Programm aus. Es sollte aussehen wie in Abbildung 10-33.

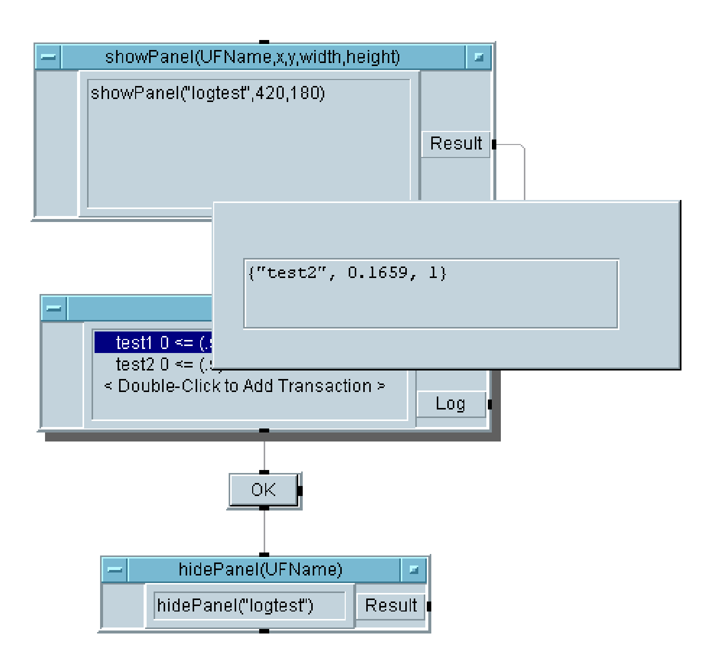


Abbildung 10-33. Das Statusanzeigeprogramm (während der Ausführung)

Das Objekt `showPanel` zeigt das `UserFunction`-Fenster an, ohne jedoch die `UserFunction` aufzurufen. Der `Sequencer` ruft die `UserFunction` zwei Mal über die Protokollfunktion auf; bei jedem dieser Aufrufe wird das Fenster aktualisiert. Wenn der Bediener mit der Ausführung fertig ist, kann er `OK` drücken, um das Fenster auszublenden. In diesem Beispiel wird ein Objekt `OK` verwendet, um das Objekt `hidePanel` auszulösen; Sie können diese Objekt jedoch auch an einer anderen Stelle im Programm platzieren, um das Timing der Ausführung zu steuern. Arbeiten Sie das Programm schrittweise ab, um das Einblenden der Statusanzeige, das Aktualisieren mit den Ergebnissen von `Test1` und `Test2` und das anschließende Ausblenden genauer zu sehen.

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich die entsprechenden Themen bei Bedarf noch einmal an, bevor Sie mit dem nächsten Kapitel fortfahren.

- Zusammenfassung der wichtigsten Punkte zu Bedienerschnittstellen.
- Verwenden eines Menüs zur Auswahl von Tests mit einer Bedienerschnittstelle.
- Importieren einer Bitmap-Datei für eine Bedienerschnittstelle.
- Erstellen einer Statusanzeige.
- Auflisten einiger von VEE bereitgestellten Funktionen der Bedienerschnittstelle.
- Sichern eines Programms.
- Erstellen einer auffälligen Warnung.
- Erstellen eines ActiveX-Steuerelements und Finden der Eigenschaften und Methoden im `Function & Object Browser`.

11

Optimieren von Agilent VEE-Programmen

Optimieren von Agilent VEE-Programmen

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Grundlegende Techniken zum Optimieren von Programmen
- Verwenden von Dynamic Link Libraries (DLLs) auf einem PC
- Optimieren mit kompilierten Funktionen
- Verwenden von kompilierten Funktionen aus anderen Sprachen auf UNIX-Plattformen
- Verwenden des VEE-Compilers
- Verwenden des VEE Profiler

Erforderliche Zeit für dieses Kapitel: 2 Stunden

Überblick

In diesem Kapitel lernen Sie, wie Sie die Ausführungsgeschwindigkeit Ihrer VEE-Programme optimieren. Bei der Leistung eines Testprogramms sind drei grundlegende Faktoren zu betrachten: die Geschwindigkeit beim Erfassen des Messwerts, die Geschwindigkeit der Übertragung von Daten an den Computer und die Geschwindigkeit, mit der das Programm die Daten verarbeitet. Durch die Optimierung des VEE-Programms können Sie seine Verarbeitungsgeschwindigkeit optimieren.

Im ersten Abschnitt lernen Sie die grundlegenden Prinzipien für die Optimierung eines VEE-Programms kennen. Außerdem lernen Sie, wie Sie Dynamic Link Libraries (DLLs) auf dem PC verwenden können. Im nächsten Abschnitt wird beschrieben, wie Sie kompilierte Funktionen optimieren können. Außerdem lernen Sie, wie Sie Teile des Programms optimieren können, indem Sie das Programm mit kompilierten Funktionen aus anderen Sprachen auf UNIX-Plattformen verbinden. Als nächstes finden Sie einen Überblick über den VEE-Compiler. Als letztes finden Sie einen Abschnitt über den VEE-Profiler.

Hinweis Die Techniken in diesem Kapitel gelten unabhängig davon, ob Sie den Compiler verwenden oder nicht.

Grundlegende Techniken zum Optimieren von Programmen

Lesen Sie zum Optimieren von VEE-Programmen die Informationen in diesem Abschnitt. Mit den hier beschriebenen Techniken können Sie sich einen guten Stil bei der Programmierung in VEE angewöhnen.

Ausführung mathematischer Operationen auf Arrays, wann immer möglich

Die Leistung eines Programms kann entscheidend verbessert werden, wenn Sie mathematische Operationen auf Arrays anwenden, wann immer dies möglich ist. Angenommen, ein Test soll die Quadratwurzel eines erfassten Messwerts ermitteln. Der herkömmliche Weg für die Programmierung dieser Aufgabe ist das Erfassen des Messwerts und die Berechnung der Wurzel in einer Schleife. In VEE können Sie stattdessen alle Messwerte in einem Array speichern und die Quadratwurzel des Arrays in einem einzigen Arbeitsschritt berechnen.

In Abbildung 11-1 durchläuft das Programm 1024 Iterationen. Jede Iteration berechnet eine Quadratwurzel.

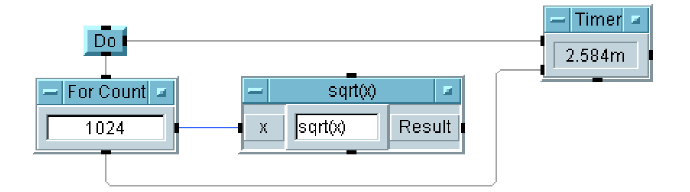


Abbildung 11-1. Berechnen von Quadratwurzeln pro Messung

In Abbildung 11-2 erstellt das Programm einen Array aus 1024 Elementen und berechnet die Quadratwurzel des Arrays (das Ergebnis ist ein Array aus Quadratwurzeln). Obwohl die beiden Programme die gleichen Ergebnisse liefern, wird das Programm in Abbildung 11-2 ca. 6 Mal schneller ausgeführt als das in Abbildung 11-1. (Bei diesem Beispiel wird von einem HP Pavilion PC mit 300 MHz ausgegangen.)

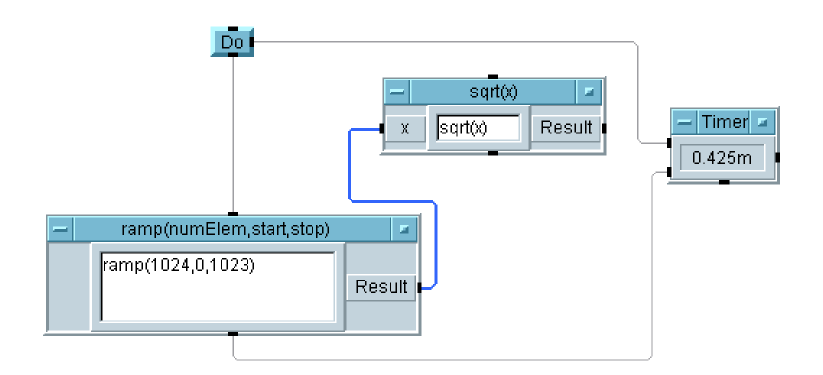


Abbildung 11-2. Berechnen von Quadratwurzeln mit Math-Array

Der Unterschied bei der Ausführungszeit der beiden Programme ergibt sich aus der erforderlichen Zeit zur Ausführung eines Objekts. Jede Ausführung eines Objekts erfordert einen bestimmten Systemaufwand. Wenn Sie daher mit Hilfe von Arrays statt skalarer Werte die Anzahl der Ausführungen für ein Objekt verringern, wird das Programm schneller ausgeführt.

Die Verwendung eines `Do`-Objekts empfiehlt sich bei der Verwendung von Timern, um sicherzustellen, dass der Timer in beiden Programmen zuerst auslöst. Die Funktion `ramp` generiert einen Array aus 1024 Elementen (von 0 bis 1023).

Hinweis

Vergewissern Sie sich für eine schnellere Ausführung, dass Sie den neuesten Ausführungsmodus in VEE verwenden. Klicken Sie hierzu `File` ⇒ `Default Preferences` an (oder verwenden Sie die Schaltfläche in der Symbolleiste). Wählen Sie `VEE6` unter `Execution Mode` aus und klicken Sie `Save` an. In der Statusleiste am unteren Rand des VEE-Fensters sollte `VEE6` aufgelistet werden.

Darstellen von Objekten als Symbole, wann immer möglich

Je mehr Informationen VEE auf dem Bildschirm verwalten muss, desto länger dauert die Ausführung eines Programms. Verwenden Sie zur Optimierung des Programms die Symboldarstellung für Objekte, die ihren Inhalt aktualisieren (z. B. der `Counter`), die Symboldarstellung statt der offenen Ansicht. Das Beispiel in Abbildung 11-3 arbeitet ca. 46 Mal schneller mit der Symboldarstellung mit den Objekten `For Count` und `Counter`.

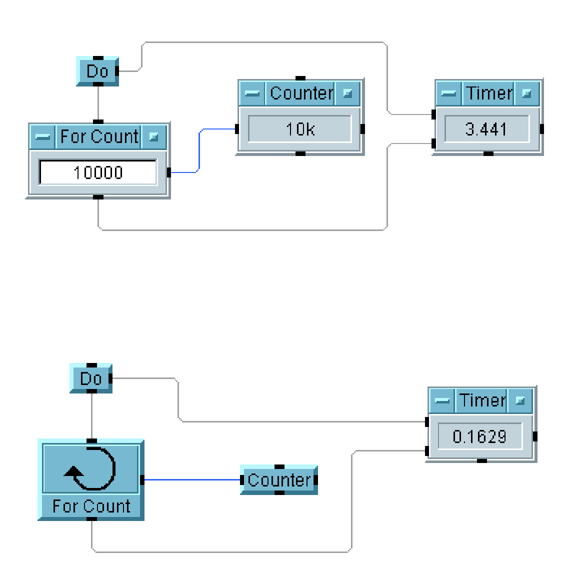


Abbildung 11-3. Optimieren von Programmen durch Verwendung von Symbolen

Reduzieren der Anzahl der Objekte in Programmen

Je mehr Erfahrung Sie sammeln, desto mehr werden Sie dazu tendieren, weniger Objekte in VEE zu verwenden. Es gibt zwei weitere Techniken zur Reduzierung der Anzahl von Objekten und somit zur Optimierung von Programmen:

1. Verwenden Sie eine einzelne Gleichung in einem `Formula`-Objekt, statt separate mathematische Objekte zu verwenden. Legen Sie beispielsweise die Gleichung `((a + b) * c)/d` in einem `Formula`-Objekt ab, statt separate Objekte für die Addition, Multiplikation und Division zu verwenden. Verwenden Sie außerdem Konstanten in der Formel statt konstanter Objekte, die mit Eingängen verbunden sind. (Legen Sie Konstanten mit `Set Variable` fest.)

2. Verschachteln Sie Funktionsaufrufe innerhalb von anderen Listen mit Funktionsparametern. In Abbildung 11-4 verwenden die Funktion randomize den von der Funktion ramp generierten Array. In Abbildung 11-5 ist der Funktionsaufruf auf `ramp` in dem Aufruf auf `randomize` verschachtelt; dies führt zu einer etwas schnelleren Programmausführung.

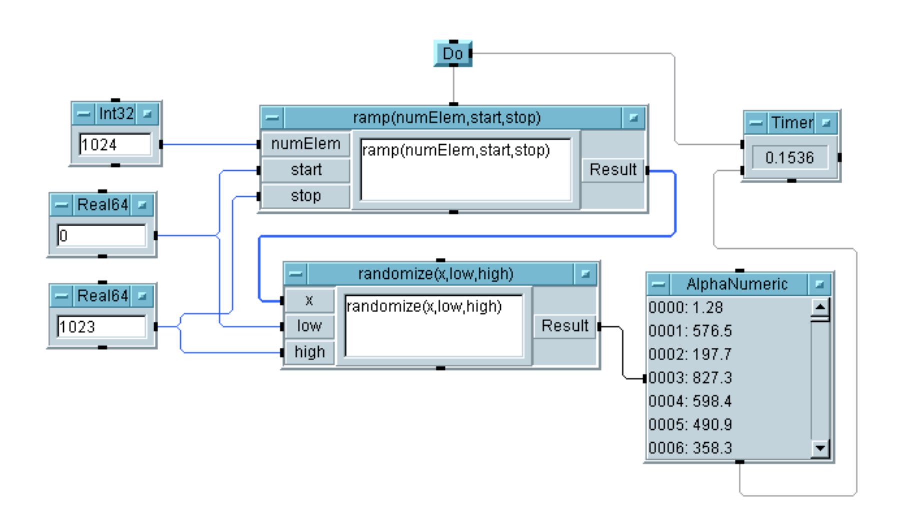


Abbildung 11-4. Funktionsaufrufe ohne Optimierung

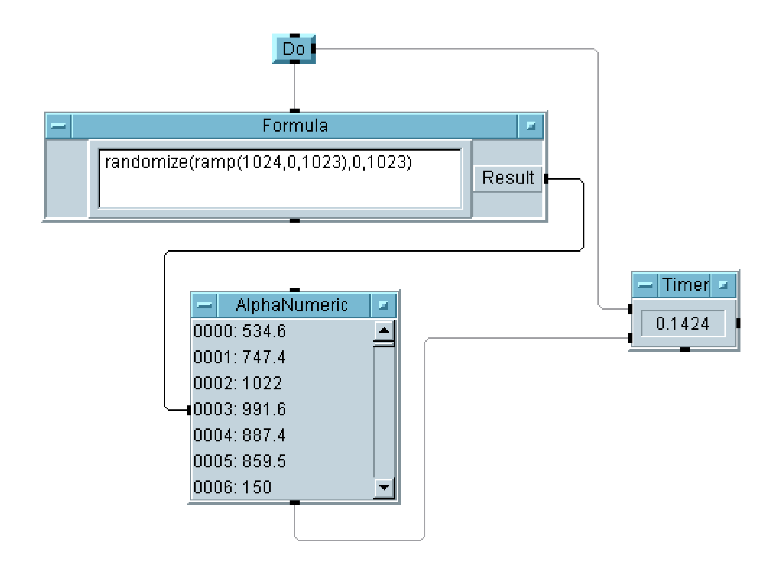


Abbildung 11-5. Funktionsaufrufe mit Optimierung

Weitere Möglichkeiten zur Optimierung von Agilent VEE-Programmen

Nachfolgend sind einige weitere Optimierungstechniken beschrieben, die Sie in Ihren Programmen anwenden können:

- Vergewissern Sie sich, dass Sie den VEE-Compiler verwenden, indem Sie Ihre Programme im Ausführungsmodus VEE 4 oder einem höheren Modus ausführen. Weitere Informationen hierzu finden Sie unter “Agilent VEE-Ausführungsmodi” auf Seite .

- Starten Sie das Programm von der Fensteransicht statt von der Detailansicht aus. VEE muss in diesem Fall weniger Objekte auf dem Bildschirm verwalten.

- Verwenden Sie globale Variablen statt der Übergabe von Werten in und aus `UserObjects` und `UserFunctions` (insbesondere für große Arrays oder Datensätze). Deklarieren Sie alle globalen Variablen. Auf diese Weise können Sie auch globale Variablen verwenden. Siehe `Data` ⇒ `Variable` ⇒ `Declare Variable`.

- Erfassen Sie Daten für grafische Anzeigen und plotten Sie den gesamten Array in einem Arbeitsschritt statt jeden Skalarpunkt einzeln. Wenn die `X`-Werte eines Plots gleichmäßige Abstände aufweisen, verwenden Sie eine Anzeige `XY Trace` statt eines Plots `X` gg. `Y`.
- Verwenden Sie ein Objekt `If/Then/Else` mit mehreren Bedingungen statt mehrerer `If/Then/Else`-Objekte.
- Definieren Sie grafische Anzeigen so einfach wie möglich. Die Einstellungen, die die schnellste Aktualisierung ermöglichen, sind `Grid Type` ⇒ `None` und keine Auswahl im Dialogfenster `Properties`. Verwenden Sie `AutoScale`-Steuer-Pins nur, wenn dies unbedingt erforderlich ist, und schalten Sie die Funktion `Automatic AutoScaling` (im Ordner `Scales`) nach Möglichkeit aus.
- Verwenden Sie beim Lesen von Daten aus einer Datei die Transaktion `ARRAY 1D TO END: (*)` statt der Ausführung von `READ`-Transaktionen mit einzelnen Elementen und der Verwendung des `EOF`-Pin.
- Verwenden Sie den `Sequencer` zum Steuern des Ausführungsablaufs mehrerer `UserFunctions` statt separater `Call`-Objekte.
- Aktivieren Sie bei Verwendung des `Sequencer` die Protokollierung für Transaktionen nur, wenn ein `Log`-Datensatz erforderlich ist.
- Wenn Sie die Anzeigen `Strip Charts` und `Logging AlphaNumeric` verwenden, stellen Sie die `Buffer Size` unter `Properties` auf den kleinsten möglichen Wert für Ihre Anwendung ein.
- Verwenden Sie den triadischen Operator *(bedingung ? ausdruck1 : ausdruck2)* statt des Objekts `If/Then/Else` mit `Gates` und einem `Junction`-Objekt
- Stellen Sie bei Verwendung von Bitmaps diese auf `Actual` oder `Centered` ein statt auf `Scaled`, da `Scaled` ein wenig länger dauert.
- Schalten Sie bei Verwendung von Indikatoren wie `Fill Bar` oder `Thermometer` die Option `Show Digital Display` aus.

- Wenn Sie mit `Color Alarms` arbeiten und schnell zwischen verschiedenen Farben wechseln müssen, schalten Sie `Show 3D Border` aus.

Zusätzlich zu den bereits erwähnten Techniken können Sie durch das Verbinden kompilierter Funktionen aus anderen Sprachen mit Ihren VEE-Programm die Ausführungsgeschwindigkeit verbessern. Die Verwendung kompilierter Funktionen auf PCs (als DLLs) ist im nächsten Abschnitt beschrieben. Die Verwendung kompilierter Funktionen auf UNIX-Plattformen ist im Abschnitt “Kompilierte Funktion mit C (UNIX)” auf Seite 448 beschrieben.

Überblick über kompilierte Funktionen

Sie können eine kompilierte Funktion, beispielsweise eine DLL (Dynamic Link Library), in einem VEE-Programm verwenden. Hierzu müssen Sie sich die kompilierte Funktion besorgen oder mit den folgenden Schritten erstellen:

1. Schreiben von Funktionen in C, C++, Fortran oder Pascal und anschließendes Kompilieren.

2. Schreiben einer Definitionsdatei für die Funktionen.

3. Erstellen einer gemeinsamen Bibliothek, die die kompilierten Funktionen enthält.

Vorteile der Verwendung kompilierter Funktionen

Die Verwendung kompilierter Funktionen in einem VEE-Programm bietet folgende Vorteile:

- Schnellere Ausführung
- Nutzung aktueller Testprogramme in anderen Sprachen
- Entwicklung von Datenfiltern in anderen Sprachen und Integration in VEE-Programmen
- Sichern eigener Routinen

Hinweis

Das Hinzufügen kompilierter Funktionen macht den Entwicklungsprozess komplexer. Verwenden Sie daher eine kompilierte Funktion *nur*, wenn der erforderliche Funktionsumfang oder die Leistung mit keinem der folgenden Elemente erzielt werden können: einer VEE-`UserFunction`, einem `Execute Program`-Ausgang zum Betriebssystem oder einem `ActiveX Automation`-Aufruf zu einem anderen Programm.

Überlegungen zum Design bei der Verwendung kompilierter Funktionen

Wenn Sie vorhaben, kompilierte Funktionen in einem VEE-Programm zu verwenden, beachten Sie die folgenden Informationen:

- Sie können alle über das Betriebssystem verfügbaren Funktionen nutzen einschließlich mathematischer Routinen, der Instrument-E/A etc. Von dem Programm aus, das Sie verbinden wollen, können Sie jedoch nicht auf die VEE-internen Funktionen zugreifen.
- Sie müssen in Ihrer kompilierten Funktion eine Fehlerprüfung bereitstellen, da VEE Fehler in einer externen Routine nicht erfassen kann.
- Sie müssen die Zuordnung des in der externen Routine zugeordneten Speichers aufheben.
- Wenn Sie Daten an eine externe Routine übergeben, vergewissern Sie sich, dass Sie die Eingangsanschlüsse des `Call`-Objekts auf den Typ und die Form der für die Routine erforderlichen Daten konfigurieren.
- System-E/A-Ressourcen werden eventuell gesperrt; Ihre externe Routine sollte diese Art von Ereignis verarbeiten können.
- Wenn Ihre externe Routine Arrays akzeptiert, muss sie einen gültigen Zeiger für die Art der untersuchten Daten aufweisen. Außerdem muss die Routine die Größe des Arrays prüfen. Wenn sich die Größe der Routine ändert, müssen Sie die neue Größe an das VEE-Programm zurückgeben.
- Die kompilierte Funktion muss als letzte Anweisung `return()` verwenden und nicht `exit()`. Wenn die kompilierte Funktion "exit" verwendet, wird auch VEE beendet, da eine kompilierte Funktion mit VEE verbunden ist.
- Wenn Sie die Grenzen eines Array überschreiben, hängt das Ergebnis von der verwendeten Sprache ab. In Pascal, das eine Prüfung der Grenzen durchführt, tritt ein Laufzeitfehler auf, und VEE wird gestoppt. In Sprachen wie C, die keine Prüfung der Grenzen durchführen, ist das Ergebnis nicht vorhersehbar; es können jedoch eine intermittierende Datenfehler auftreten oder ein Absturz von VEE.

Richtlinien zur Verwendung kompilierter Funktionen

Beachten Sie bei Verwendung kompilierter Funktionen in einem VEE-Programm die folgenden Richtlinien:

- Sie können eine `kompilierte Funktion` genau wie eine `UserFunction` aufrufen und kompilieren. Sie können die gewünschte Funktion entweder mit `Select Function` über das `Call`-Objektmenü auswählen oder ihren Namen eingeben. In beiden Fällen werden die Ein- und Ausgangsanschlüsse des Objekts `Call Function` automatisch konfiguriert, sofern VEE die Funktion erkennt. Die erforderlichen Informationen werden über die Definitionsdatei bereitgestellt. (VEE erkennt, ob die Bibliothek bereits importiert wurde.)

- Zum Umkonfigurieren der Ein- und Ausgangsanschlüsse von `Call` wählen Sie `Configure Pinout` im Objektmenü aus. Bei beiden Methoden konfiguriert VEE das `Call`-Objekt mit den für die Funktion erforderlichen Eingangsanschlüssen und mit einem `Ret Value`-Ausgangsanschluss für den Rückgabewert der Funktion. Darüber hinaus gibt es einen Ausgangsanschluss zu jedem als Referenz übergebenen Eingang.

- Rufen Sie die `kompilierte Funktion` von einem Ausdruck in einem `Formula`-Objekt oder von anderen Ausdrücken aus, die zur Ausführungszeit ausgewertet werden, mit ihrem Namen auf. Sie könnten eine `kompilierte Funktion` auch aufrufen, indem Sie ihren Namen in einem Ausdruck innerhalb einer `To File`-Transaktion angeben.

Hinweis

Lediglich der Rückgabewert der `kompilierten Funktion` (`Ret Value` im `Call`-Objekt) kann von innerhalb eines Ausdrucks abgerufen werden. Wenn Sie weitere von der Funktion zurückgegebene Parameter abrufen wollen, müssen Sie das Objekt `Call` verwenden.

- Löschen Sie eine Bibliothek von `kompilierte Funktionen` durch Verwenden des Objekts `Delete Library` im Menü `Device`. Mit den Objekten `Import Library`, `Call` und `Delete Library` können Sie die Ladezeit des Programms verkürzen und Speicherplatz sparen, indem Sie die Objekte importieren und wieder löschen, wenn das Programm den Aufruf dieser Objekte abgeschlossen hat.

Verwenden der Dynamic Link Libraries

Auf PCs können Sie die kompilierten Funktionen aus Dynamic Link Libraries (DLLs) als Teil eines VEE- Programms verwenden. DLLs können kompilierte Funktionen sein, die Sie selbst geschrieben haben oder DLLs, die Sie gekauft oder über das Web heruntergeladen haben. (Informationen zum Schreiben eigener DLLs können Sie über Microsoft anfordern.)

Hinweis

VEE unterstützt die beiden Aufrufkonventionen "`_cdecl`" und "`_stdcall`". Die meisten von Kunden selbst geschriebenen DLLs verwenden die Aufrufkonvention `_cdecl`. Die meisten Win32 API-Aufrufe verwenden `_stdcall`. VEE unterstützt beide Namenskonventionen; Sie können somit die meisten gängigen DLLs wie auch eigene DLLs verwenden.

Integrieren einer DLL in ein Agilent VEE-Programm

In diesem Abschnitt wird beschrieben, wie Sie eine DLL in ein VEE-Programm importieren. Schreiben oder besorgen Sie sich die DLL wie oben beschrieben, und führen Sie zur Verwendung der DLL die folgenden Schritte durch:

1. Wählen Sie `Device ⇒ Import Library` aus.

   Der `Library Type` bietet drei Auswahlmöglichkeiten: `UserFunction`, `Compiled Function` und `Remote Function`. Ändern Sie den Eintrag im Feld `Library Type` in `Compiled Function` (der Standardwert lautet `UserFunction`). Für eine `Compiled Function` enthält das Objekt `Import Library` ein Feld für die `Definition File` (siehe Abbildung 11-6).

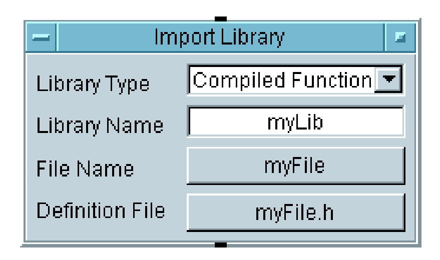


Abbildung 11-6. Importieren einer Bibliothek kompilierter Funktionen

Die Felder sind nachfolgend beschrieben:

Library Name (Name der Bibliothek)	Der Name, über den VEE die Bibliothek identifiziert. Im allgemeinen wird dieser Name verwendet, wenn Sie die Bibliothek löschen wollen, nachdem Sie im Programm verwendet wurde.
File name (Dateiname)	Die Datei, die die gemeinsame Bibliothek enthält.
Definition File (Definitionsdatei)	Die Include-Datei mit den Prototypen der Funktionen. Dies ist normalerweise eine `*.h`-Datei.

Hinweis Sie können auch in der Entwicklungsphase eine Bibliothek manuell laden, indem Sie `Load Lib` im Objektmenü auswählen.

2. Wählen Sie `Device` ⇒ `Call` aus.

Wenn Sie die Bibliothek mit `Import Library` importiert haben, erstellen Sie ein `Call`-Objekt durch Auswählen von `Device` ⇒ `Call`. Sie können anschließend die `kompilierte Funktion` aufrufen, indem Sie `Select Function` im `Call`-Objektmenü und die gewünschte Funktion im angezeigten Listenfeld auswählen. Das in Abbildung 11-7 dargestellte `Call`-Objekt ruft beispielsweise die `kompilierte Funktion` mit dem Namen `myFunction` in `myLibrary` mit den Parametern `arraySize` und `array` auf.

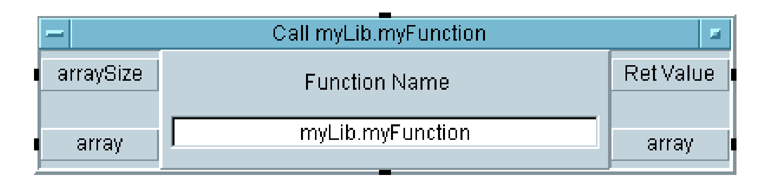


Abbildung 11-7. Verwenden von Call-Objekten für kompilierte Funktionen

VEE konfiguriert automatisch das `Call`-Objekt mit dem Funktionsnamen und der richtigen Anzahl von Ein- und Ausgangs-Pins. Ein zweiter, dritter usw. Pin wird für alle Parameter zugeordnet, die als Referenz an die Funktion übergeben werden. Wenn Sie den Funktionsnamen eingegeben haben, können Sie das Objekt auch konfigurieren, indem Sie `Configure Pinout` im Objektmenü auswählen.

Hinweis

Sie können auch eine DLL-Funktion von einem Ausdrucksfeld aus aufrufen, sofern die Bibliothek geladen wurde. Bei einer Verwendung auf diese Weise müssen Sie die Parameter in Klammern nach dem Funktionsnamen eingeben; die Funktion sendet dann nur ihren Rückgabewert zurück. Alle als Referenz übergebenen Parameter können nur über das Objekt `Call` abgerufen werden. Sie können beispielsweise den folgenden Ausdruck in einem `Formula`-Objekt verwenden:

```
2 * yourFunc(a,b)
```

Die Angaben `a` und `b` beziehen sich auf die beiden Eingangs-Pins am `Formula`-Objekt, und der Rückgabewert von `yourFunc` wird mit `2` multipliziert und am Ausgangs-Pin angelegt.

3. (Optional) Klicken Sie `Device` ⇒ `Delete Library` an.

   Bei der Entwicklung des Programms können Sie auch `Delete Lib` im Objektmenü auswählen, um die Bibliothek programmgesteuert zu löschen. Durch das Löschen der Bibliothek nach ihrer Verwendung im Programm wird die Ladezeit verringert und Speicherplatz gespart.

Ein Beispiel zur Verwendung einer DLL

In dieser Übung importieren Sie eine DLL und rufen eine Funktion von dieser DLL aus auf. Die verwendete DLL ist im VEE-Produkt unter Windows enthalten. (Dieses Programm ist für die Konzeption auf allen Plattformen konzipiert.)

Öffnen Sie die Datei `manual49.vee`. Sie befindet sich unter:

<*installation directory*>`\EXAMPLES\MANUAL\MANUAL49.`

Sehen Sie sich dieses Beispiel genau an. Es sollte aussehen wie in Abbildung 11-8.

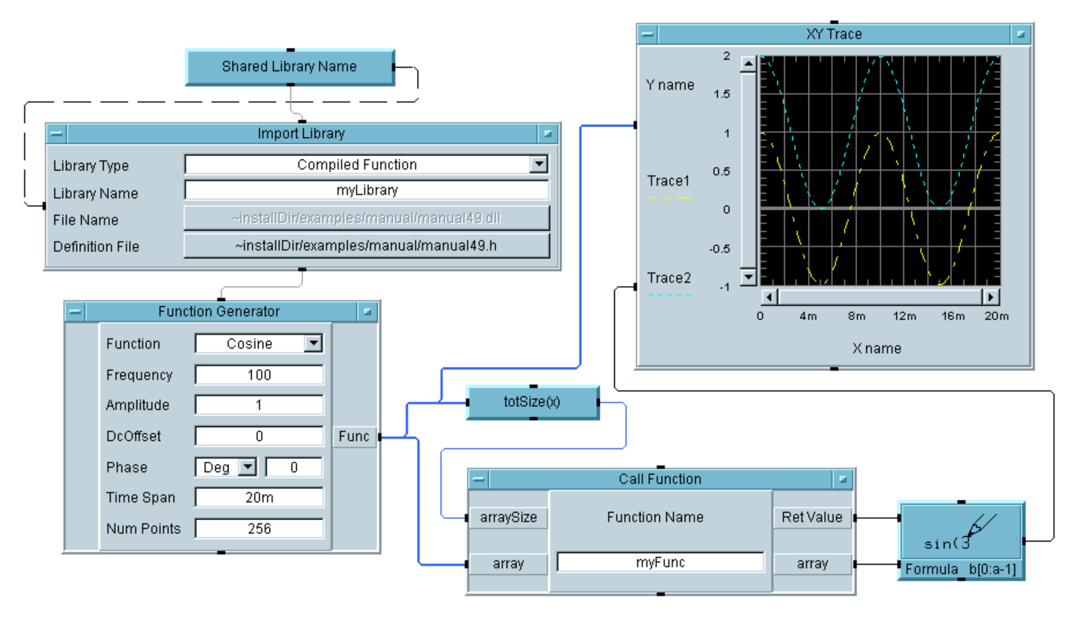


Abbildung 11-8. Ein Programm, das eine DLL (MANUAL49) verwendet

Import Library Vor dem ersten Aufruf der kompilierten Funktion `Call Function` muss die DLL mit dem Objekt `Import Library` (im Menü `Device`) geladen werden.

Call Function `MANUAL49` ruft eine kompilierte Funktion `myFunc` auf. `MyFunc` erfordert den C-Datentyp `long`; dieser Typ entspricht dem VEE-Datentyp `Int32`. Diese Zahl gibt die Größe eines Array an. Der zweite Parameter ist ein Zeiger auf den Array aus Real-Zahlen. Die Definitionsdatei befindet sich in `MANUAL49.H`, und die Quellendatei für den -Code befindet sich in `MANUAL49.C`. `MyFunc` addiert `1` zu jedem Element des Array.

Function Generator Der `Function Generator` wird zum Erstellen einer Wellenform verwendet, die den Ausgang zum Array-Pin am Objekt `Call myFunc` darstellt.

totSize Mit dem Objekt `totSize` (im Fenster `Math & Functions` box) wird die Größe der Wellenform ermittelt; diese Größe ist der Ausgang zum `arraySize`-Eingangs-Pin von `Call myFunc`.

XY Trace Das Objekt `XY Trace` zeigt die ursprüngliche und die neue Wellenform an.

Formula Der Ausgangs-Pin `Ret Value` des `Call`-Objekts enthält die Größe des zurückgegebenen Arrays, sodass der Ausdruck `B[0:A-1]` in dem `Formula`-Objekt diesen Array für das Anzeigeobjekt richtig angibt.

Führen Sie das Programm aus und beachten Sie, dass der zweite Trace an allen Punkten der Wellenform um eins größer ist als der erste Trace.

Ein weiterer wichtiger Punkt ist die Methode, mit der das Programm für alle VEE-Plattformen portierbar gemacht wird. Die HP-UX-Plattformen verwenden gemeinsame Bibliotheken, die durch die Namenserweiterung `*.sl` gekennzeichnet sind. Windows 95, Windows 98, Windows 2000 und Windows NT 4.0 verwenden einen Microsoft 32-Bit-Compiler. Alle diese DLLs sind durch die Erweiterung `*.dll` gekennzeichnet.

Das UserObject mit dem Namen Shared Library Name kennzeichnet das verwendete Betriebssystem und überträgt den richtigen Bibliotheksnamen an das Objekt Import Library, wie in Abbildung 11-9 gezeigt.

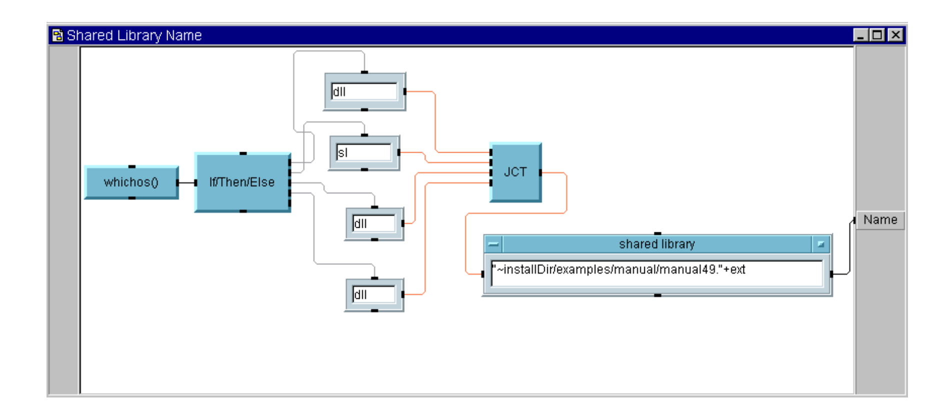


Abbildung 11-9. Das UserObject "Shared Library Name"

Die Funktion whichos() wurde in einem umbenannten Formula-Objekt zur Kennzeichnung des Betriebssystems verwendet. Ein erweitertes If/Then/Else-Objekt untersucht die Ausgabe der Funktion whichos() und löst anschließend die entsprechende Textkonstante aus. Diese Dateierweiterung wird anschließend mit einem umbenannten Formula-Objekt an den Dateinamen MANUAL49 angehängt. (Der Eingangsanschluss am Formula-Objekt mit dem Namen shared library wurde ebenfalls in ext umbenannt.)

Ein Steuer-Pin für einen File Name wurde dem Objekt Import Library hinzugefügt; daher wurde eine gepunktete Linie zwischen dem UserObject und der Import Library gezogen.

Hinweis

Überprüfen Sie To/From Socket auf die gemeinsame Verwendung von Daten in Mischumgebungen, z. B. beim Senden von Daten an eine Datenbank. Auch die Beispielprogramme "Battleship" und "Euchre" machen intensiven Gebrauch von Sockets bei der Kommunikation zwischen verschiedenen VEE-Prozessen.

Das Objekt "Execute Program" im Vergleich zu kompilierten Funktionen

Wenn Sie entscheiden, ob Sie ein Objekt `Execute Program` oder eine kompilierte Funktion zum Integrieren Ihrer kompilierten Programme mit VEE verwenden wollen, beachten Sie die folgenden Informationen.

Das Objekt "Execute Program"

Das Objekt `Execute Program` zeichnet sich durch folgende Merkmale aus:

- Einfache Bedienung
- Längere Startzeit
- Kommunikation über Pipes (nur HP-UX)
- Geschützter Adressbereich
- Auswahl zwischen synchroner und asynchroner Ausführung
- Service der asynchronen Ereignisse
- Hohe Sicherheit (wenn das aufgerufene Programm abstürzt, erhalten Sie eine Fehlermeldung)
- Besser geeignet für eine laufende Datenerfassung

Kompilierte Funktionen

Kompilierte Funktionen, die von den Objekten `Import Library` und `Call` verwendet werden, zeichnen sich durch folgende Merkmale aus:

- Kürzere Startzeit
- Kommunikation durch Weitergabe von Stack- und Speicherbereichen, die mit VEE gemeinsam genutzt werden
- Synchrone Ausführung
- Signale und Ausnahmebedingungen werden nicht blockiert oder erfasst (z. B. "Allgemeine Schutzverletzungen")

- Compiler für Textsprachen erforderlich
- Kompliziertere Bedienung. Höheres Risiko bei der Verwendung ("Out-of-bounds"-Array-Fehler oder ein Überschreiben des Speichers führt zu einem Absturz von VEE)

Kompilierte Funktion mit C (UNIX)

Der Prozess der Verwendung kompilierter Funktionen aus anderen Sprachen erfordert gemeinsame Bibliotheken in HP-UX-Plattformen. Sie können ein in C, C++, Fortran oder Pascal geschriebenes Programm dynamisch mit einem VEE-Programm auf einer UNIX-Workstation verbinden ("link"). Beachten Sie, dass kompilierte Pascal-Funktionen nur auf den HP 9000 Workstations der Serie 700 unterstützt werden.

Das in Abbildung 11-10 dargestellte Programm importiert eine Bibliothek und ruft eine C-Funktion auf. Die C-Funktion akzeptiert einen Real-Array und addiert zu jedem Element in dem Array die Zahl 1. Der geänderte Array wird an VEE über den `Array`-Anschluss des Objekts `Call Function` zurückgegeben, und die Größe des Arrays wird am Anschluss `Ret Value` zurückgegeben. Dieses Beispiel befindet sich im folgenden VEE-Verzeichnis:

<*Installationsverzeichnis*>`/examples/manual/manual49.vee.`

Hinweis

Die Dateierweiterungen lauten wie folgt: Eine Erweiterung “`.vee`” kennzeichnet ein Programm, eine Erweiterung “`.c`” kennzeichnet eine Datei mit C-Quellencode, eine Erweiterung “`.h`” oder “`.def`” kennzeichnet eine Definitionsdatei, und eine Erweiterung “`.sl`” kennzeichnet eine gemeinsame Bibliotheksdatei.

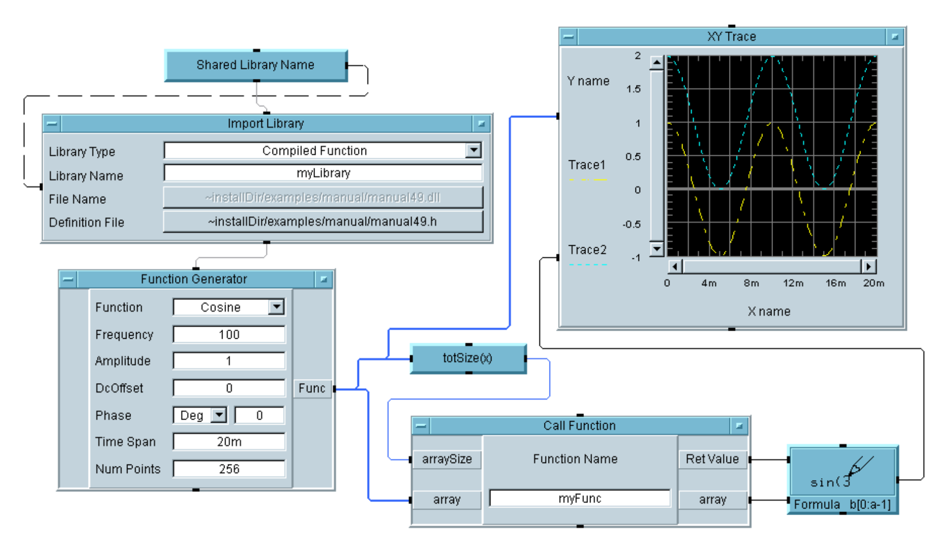


Abbildung 11-10. Programm mit Aufruf einer kompilierten Funktion

Beachten Sie folgende Hinweise zu dem Programm:

Größe des Arrays

Eine Variable in der C-Funktion (und entsprechend ein Dateneingangsanschluss im `Call`-Objekt) wird zur Angabe der Größe des Arrays verwendet. Mit der Variablen `arraySize` wird verhindert, dass Daten über das Ende des Arrays hinaus überschrieben werden.

Das Objekt "Call Function"

Da der Array als Referenz übergeben wurde, erstellt VEE automatisch einen Eingangs- und einen Ausgangs-Pin am Objekt `Call Function`.

Die Variable `arraySize` wurde als Wert übergeben, daher erstellt VEE lediglich einen Eingangsanschluss. Der Rückgabewert der Funktion wird jedoch zur Rückgabe der Größe des Ausgabe-Arrays an VEE verwendet. Diese Technik ist nützlich, wenn Sie einen Array zurückgeben müssen, der weniger Elemente als der Eingabe-Array hat.

Die C-Routine Die C-Routine ist eine Funktion und nicht eine Prozedur. Die `kompilierte Funktion` erfordert einen Rückgabewert; wenn Sie daher eine Sprache verwenden, die zwischen Prozeduren und Funktionen unterscheidet, vergewissern Sie sich, dass Sie die Routine als Funktion schreiben.

Reihenfolge der Ausführung Das Objekt `Import Library` wird im Programm vor dem `Call`-Objekt ausgeführt. Wenn Sie sich bei der Reihenfolge der Ausführung für diese beiden Objekte nicht sicher sind, verwenden Sie die Sequenz-Pins, um die richtige Reihenfolge sicherzustellen.

Übergeben von Variablen Die als Referenz an die Funktion übergebene Parametervariable "Array" hat sowohl Eingangs- als auch Ausgangsanschlüsse; die als Wert übergebene Variable `arraySize` hat jedoch nur einen Eingangsanschluss.

Anzahl der Array-Elemente Das `Formula`-Objekt verwendet die Größe des Arrays im Anschluss `Ret Value` zum Senden der richtigen Anzahl von Array-Elementen an die Anzeige.

XY Trace Der `XY Trace` skaliert die beiden Wellenformen automatisch.

Agilent VEE-Ausführungsmodi

Die Agilent VEE-Ausführungsmodi ermöglichen die Ausführung von Programmen, die mit älteren Versionen von VEE erstellt wurden. Mithilfe der Ausführungsmodi können mit einer neueren Version von VEE auch Programme ausgeführt werden, die mit einer älteren Version von VEE geschrieben wurden, und zwar in exakt der gleichen Weise wie mit der älteren Version von VEE. Dadurch ist VEE rückwärtskompatibel und unterstützt bestehende Programme.

Hinweis

Der *Ausführungsmodus* wurde in älteren Versionen von VEE als *Kompatibilitätsmodus* bezeichnet.

VEE kennt vier Ausführungsmodi:

- VEE 6 (fügt neue Datentypen hinzu)
- VEE 5 (fügt ActiveX hinzu)
- VEE 4 (kompiliert)
- VEE 3.*x*

Der Ausführungsmodus des ausgeführten Programms wird in der Statusleiste von VEE angezeigt, wie in Abbildung 11-11 dargestellt.

Wenn Sie ein bestehendes Programm in VEE ausführen, wird standardmäßig der Ausführungsmodus für die VEE-Version verwendet, in der das Programm erstellt wurde. Ein VEE 5.0-Programm, das Sie in VEE 6.0 öffnen, wird beispielsweise standardmäßig im VEE 5 Ausführungsmodus ausgeführt.

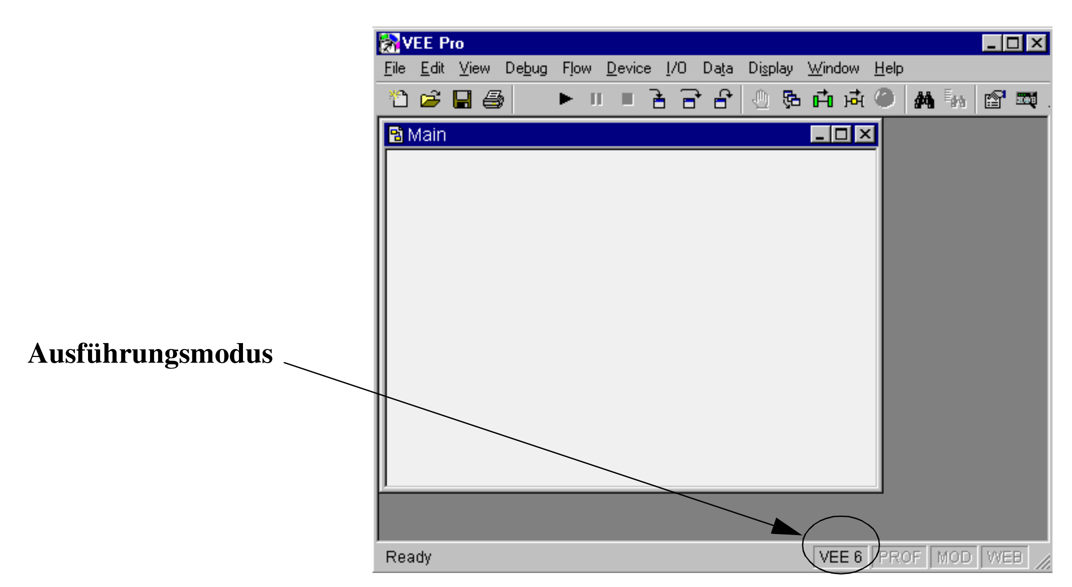


Abbildung 11-11. Anzeige des Ausführungsmodus in der VEE-Statusleiste

Der Agilent VEE-Compiler

Der VEE-Compiler wird in VEE 4 und höheren Ausführungsmodi automatisch aktiviert. Der Compiler ermöglicht eine weitaus schnellere Programmausführung sowie eine besser vorhersehbare Weitergabe von Objekten. Weitere Informationen zum Compiler und zu den Unterschieden zwischen den verschiedenen Ausführungsmodi finden Sie im Handbuch *VEE Pro Advanced Techniques*.

Ändern des Ausführungsmodus

Sie sollten alle neuen Programme im VEE 6-Modus erstellen. Falls Sie bestehende Programme haben, empfiehlt es sich, den Ausführungsmodus zu ändern, wenn Sie einem bestehenden Programm neue Funktionen hinzufügen. Wenn Sie beispielsweise ein Programm in VEE 5.0 geschrieben haben und eine neue Funktion aus VEE 6.0 hinzufügen, sollten Sie den Ausfüh-

rungsmodus in VEE 6 ändern; andernfalls wird das VEE 5.0-Programm eventuell nicht richtig ausgeführt.

Führen Sie zum Ändern des Ausführungsmodus die folgenden Schritte aus:

1. Klicken Sie im Hauptmenü von VEE die Auswahl `File ⇒ Default Preferences` an oder verwenden Sie die Schaltfläche **Standardeinstellungen** in der Symbolleiste (siehe Abbildung 11-12).

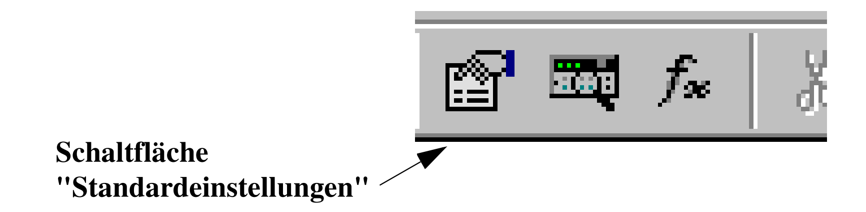


Abbildung 11-12. Schaltfläche "Standardeinstellungen" in der Symbolleiste

2. Wählen Sie im Ordner `General` (wird als oberster Ordner bereits angezeigt) unter `Execution Mode` die Option VEE 6.0 aus (siehe Abbildung 11-13). Vergewissern Sie sich im gleichen Ordner, dass die Option `Disable Debug Features` *nicht* ausgewählt ist. Klicken Sie `OK` an.

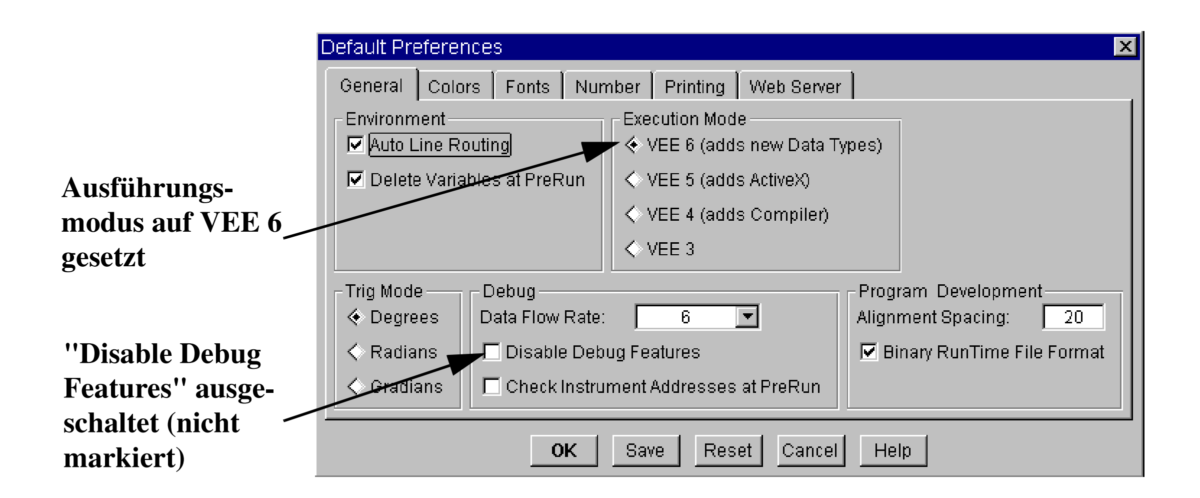


Abbildung 11-13. Ändern des Ausführungsmodus unter "Default Preferences"

Die Auswirkungen der Änderung des Ausführungs-modus

Das folgende Beispiel zeigt die Steigerung der Geschwindigkeit bei der Aktualisierung eines Programms. Bei diesem Beispiel steht die Geschwindigkeit des Programms ohne die E/A-Geschwindigkeit des Geräts im Vordergrund.

1. Öffnen Sie das Programm `chaos.vee` im Unterverzeichnis `examples\Applications`.

   Dieses Programm illustriert das explosionsartige Bevölkerungswachstum. Sie können das Programm wie hier gezeigt ändern, indem Sie die Ergebnisse mit einem `Timer`-Objekt prüfen. Diese Beispiele wurden auf einem 300MHz HP Pavillion PC unter Windows 95 ausgeführt, wobei zwei weitere große Anwendungen gleichzeitig aktiv waren.

In Abbildung 11-14 erfolgt das Timing der Programmausführung mit den im VEE 3-Ausführungsmodus geöffneten Anzeigen.

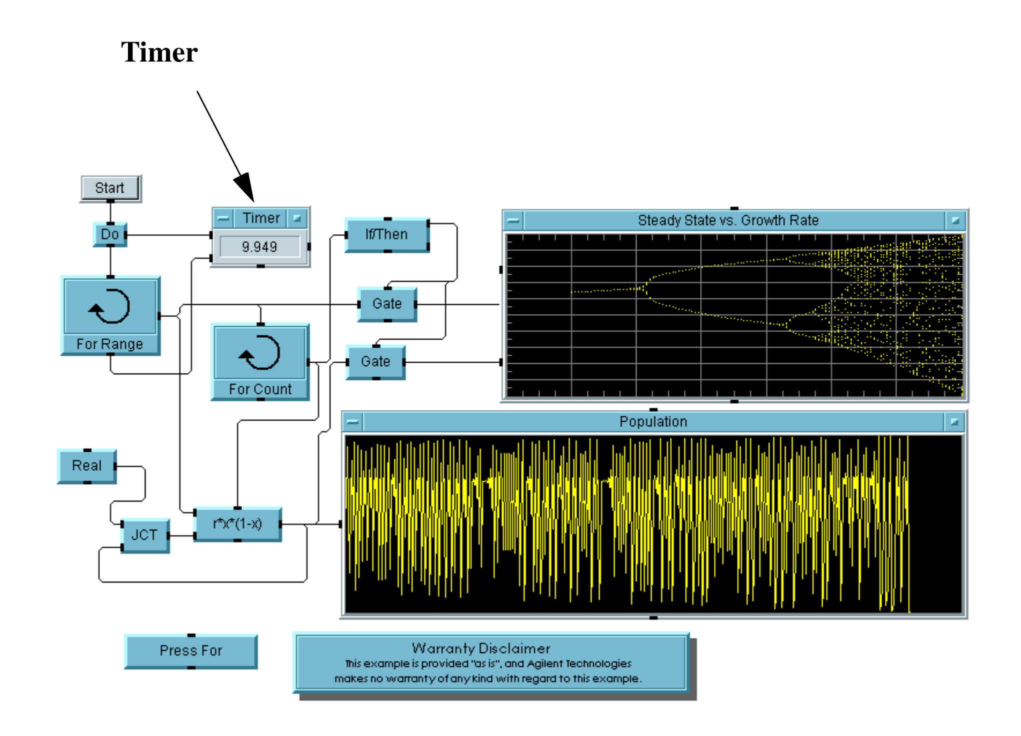


Abbildung 11-14. Chaos.vee im VEE 3-Modus mit geöffneten Anzeigen

In Abbildung 11-15 sind die Anzeigen als Symbol dargestellt, um die Geschwindigkeit zu verbessern, ohne den Compiler einzuschalten. Die Ausführungszeit wird dadurch um ca. 1/6 verkürzt.

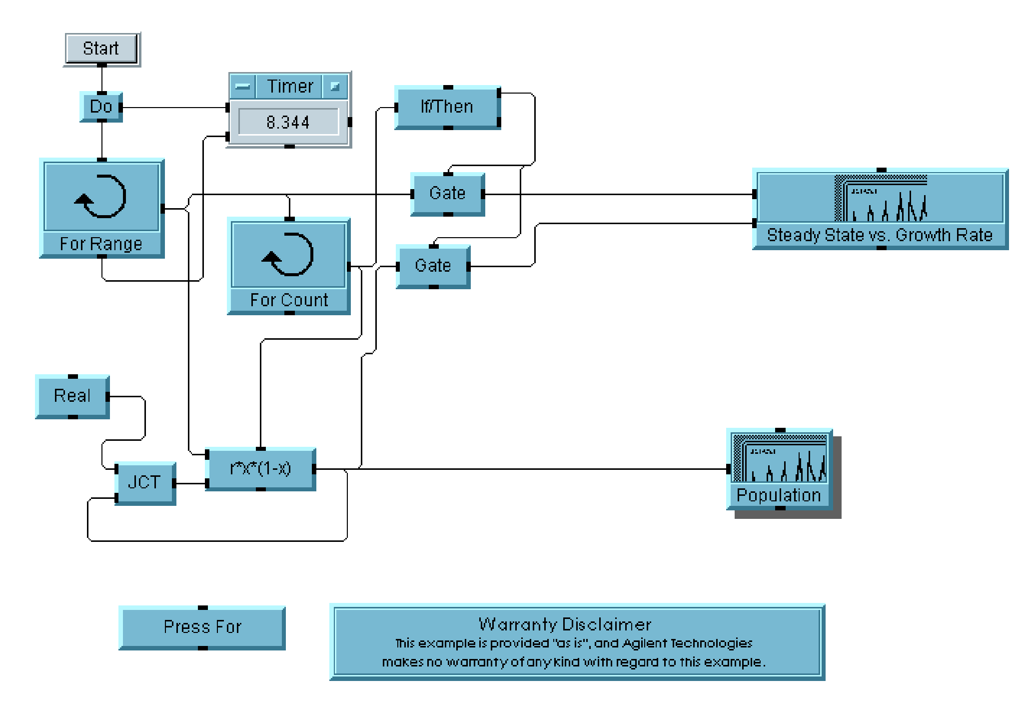


Abbildung 11-15. Chaos.vee im VEE 3-Modus mit geschlossenen Anzeigen

In Abbildung 11-16 schließlich wurden der Compiler eingeschaltet und die Fehlerbehebungsfunktionen ("debugging") ausgeschaltet. Für eine optimale Leistung markieren Sie das Feld `Disable Debug Features` unter `File` ⇒ `Default Preferences`, wenn das Programm bereits bereinigt wurde und einsatzbereit ist.

Die Funktionen zur Fehlerbehebung aktivieren bestimmte Tools einschließlich `Show Execution Flow` und `Activate Breakpoints`. Wenn Sie `Disable Debug Features` markieren, werden dadurch die Größe (im Speicher) und die Geschwindigkeit des Programms verbessert. Wie Sie sehen, läuft das Programm jetzt ca. 12 mal schneller ab. Diese drei Zahlen zeigen, wie Sie durch Kombinieren der Optimierungstechniken mit dem Compiler die beste Ausführungsgeschwindigkeit für Ihr Programm erzielen.

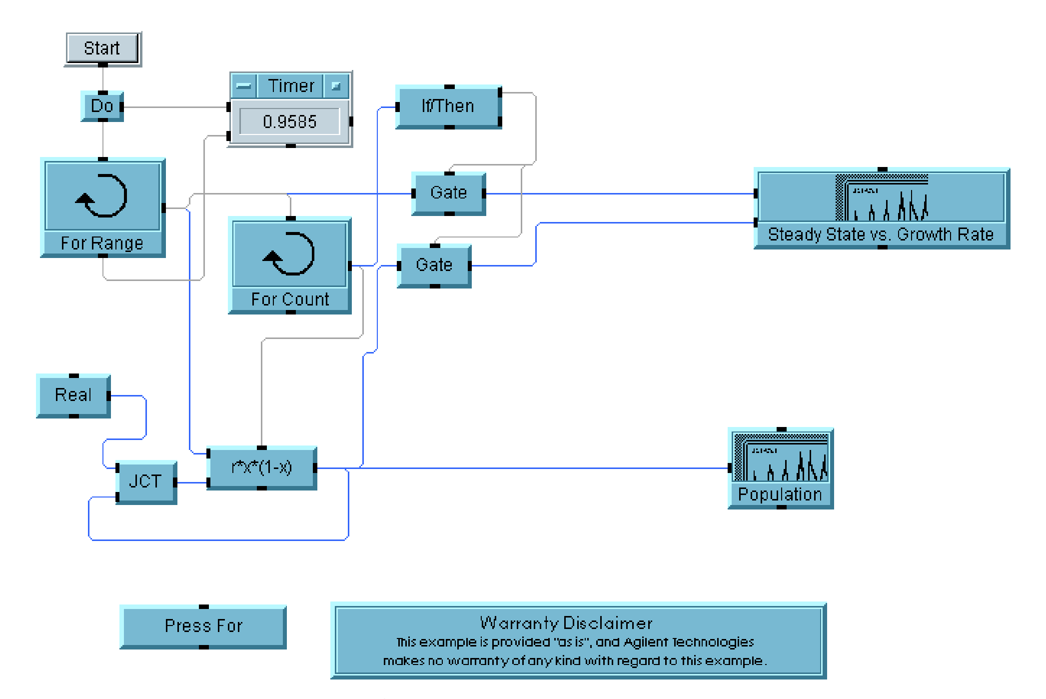


Abbildung 11-16. Chaos.vee in VEE 4 oder einem höheren Modus mit ausgeschalteter Fehlerbehebung

In Abbildung 11-17 und Abbildung 11-18 verwendet VEE den Compiler zur Verbesserung der Geschwindigkeit in Bereichen des Programms, in denen iterative skalare mathematische Routinen zum Einsatz kommen. Das Beispiel berechnet die Quadratwurzel eines skalaren Werts. (Das Ergebnis wird nicht gespeichert.) Durch die Verwendung des Compilers ist die Geschwindigkeit ca. 107 mal höher als bei Verwendung des VEE 3-Ausführungsmodus.

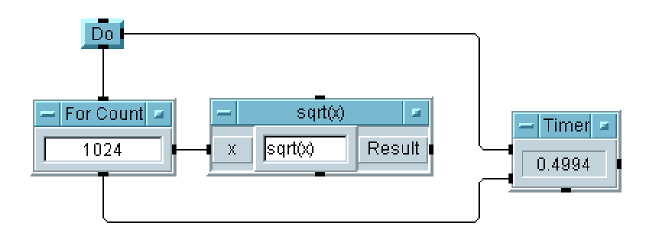


Abbildung 11-17. Iteratives mathematisches Beispiel im VEE 3-Modus

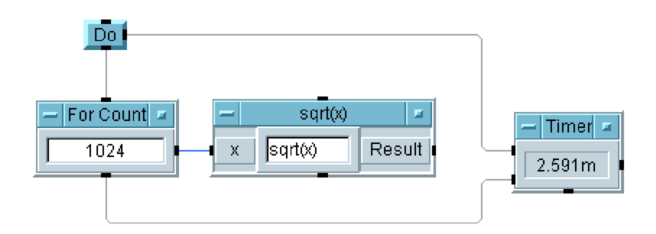


Abbildung 11-18. Iteratives mathematisches Beispiel mit VEE 4- oder einem höheren Modus

VEE stellt die Ausführungsmodi zur Verfügung, weil es in älteren Versionen von VEE einige Programmieroptionen gab, die in der aktuellen Version nicht zulässig sind (sie generieren jetzt eine Fehlermeldung). Außerdem haben die Erweiterungen bei der ActiveX-Automatisierung und den Steuerelementen dazu geführt, dass einige Programme, die in VEE 4 oder VEE 5 ausgeführt werden konnten, für die Ausführung im `VEE 6`-Modus einige kleine Änderungen erfordern. Ausführliche Informationen zu den Unterschieden zwischen den verschiedenen Ausführungsmodi finden Sie im Handbuch *VEE Pro Advanced Techniques*. Bei allen neuen Programmen sollten Sie im VEE 6- Modus beginnen.

Der Agilent VEE Profiler

Der `Profiler` ist eine Funktion in der Professional Development Environment in VEE. Der `Profiler` erleichtert die Optimierung von Programmen durch die Anzeige der Ausführungsgeschwindigkeiten von `UserFunctions` oder `UserObjects` im Programm.

Sie können mit dem `Profiler` die langsamen Stellen in einem Programm ermitteln und die in diesem Kapitel beschriebenen Techniken anwenden, um die Geschwindigkeit des Programms zu verbessern. Abbildung 11-19 zeigt das Programm `examples\Applications\mfgtest.vee`.

Zum Einschalten des Profiler wählen Sie `View` ⇒ `Profiler` aus. Führen Sie anschließend das Programm aus. Sie sehen den `Profiler` in der unteren Hälfte des Bildschirms. Der `Profiler` listet vergleichende Informationen zu der erforderlichen Ausführungsdauer der einzelnen `UserObjects` und `UserFunctions` auf.

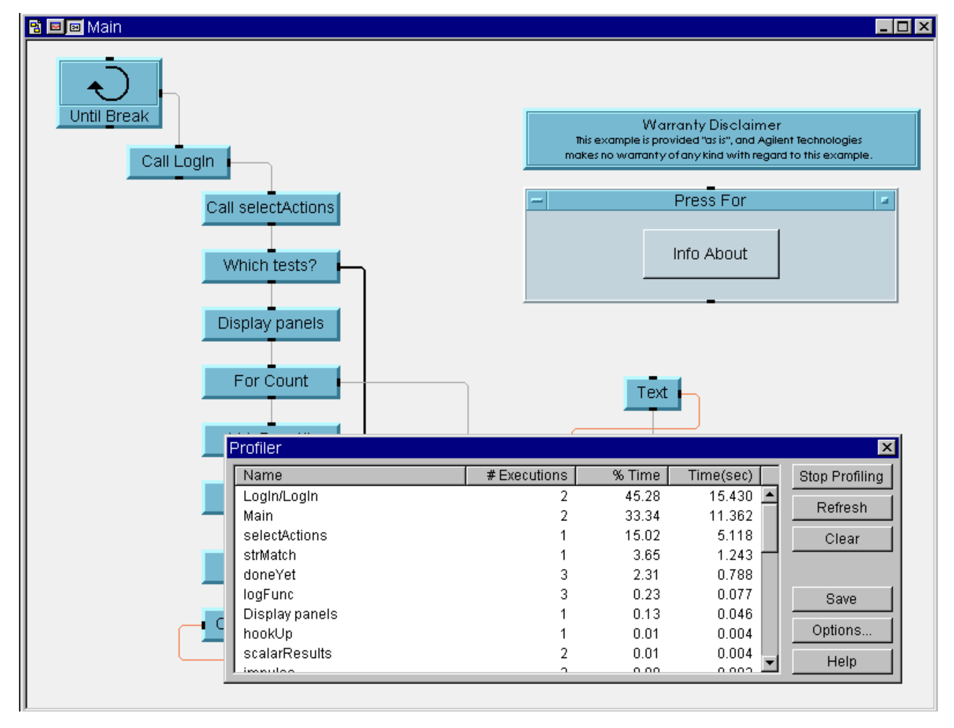


Abbildung 11-19. Ein Beispiel zum Profiler

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen. Sehen Sie sich ggf. die entsprechenden Themen noch einmal an.

- Erläutern von drei grundlegenden Techniken zur Optimierung von VEE-Programmen mit Beispielen.
- Erläutern von mindestens zwei weiteren Techniken.
- Erläutern des Grundkonzepts einer DLL.
- Importieren einer DLL, Aufrufen einer darin enthaltenen Funktion und Löschen der DLL.
- Erläutern der Verwendung einer kompilierten Funktion in einer anderen Sprache in einer HP-UX-Plattform.
- Schrittweise Ausführung eines Programms mit dem VEE 6-Ausführungsmodus oder den VEE 5-, VEE 4- oder VEE3-Ausführungsmodi sowie Erläutern der Gründe zur Verwendung der verschiedenen Modi.
- Verwenden des VEE Profiler.

12

Plattformspezifische Faktoren und Web-Überwachung

Plattformspezifische Faktoren und Web-Überwachung

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Unterschiede zwischen PC- HP-UX-Plattformen
- Kommunikation mit einem Rocky Mountain Basic-Programm
- Aufruf von VEE-Funktionen aus anderen Anwendungen mit dem VEE ActiveX-Automatisierungs-Server
- Web-Überwachung

Erforderliche Zeit für dieses Kapitel: 2 Stunden

Überblick

In diesem Kapitel lernen Sie die wichtigsten Unterschiede zwischen den Betriebssystemen kennen und zu der Art, wie VEE-Objekte diese Unterschiede verarbeitet. Anschließend erfahren Sie mehr über die wichtigsten Techniken zum Einbinden von VEE-Funktionen in andere Anwendungen und Programme mit dem aufrufbaren VEE ActiveX-Automatisierungs-Server. Zum Abschluss lernen Sie die Konzepte zur Web-Überwachung kennen.

VEE-Programme können zwischen den unterstützten Plattformen ausgetauscht werden; es gibt jedoch einige Objekte, die nur für bestimmte Betriebssysteme gültig sind. Auf einem PC verwendet VEE beispielsweise Dynamic Link Libraries (DLLs) als kompilierte Funktionen, auf HP-UX dagegen gemeinsame Bibliotheken. VEE verwendet die ActiveX-Automatisierung auf einem PC für die prozessübergreifende Kommunikation und benannte Pipes unter HP-UX.

Unterschiedene zwischen PC- und HP-UX-Plattformen

Es gibt mehrere Unterschiede zwischen der Verwendung von VEE auf einem PC und unter HP-UX.

Programme

Alle VEE-Funktionen können in Programmen auf allen Plattformen verwendet werden. Wenn das Objekt jedoch von dem verwendeten Betriebssystem abhängt, kann es nur unter diesem Betriebssystem ausgeführt werden. Sie können diese Unterschiede beim Portieren von Programmen berücksichtigen, indem Sie die Objekte `System Information` im Fenster `Function & Object Browser` verwenden: `whichOS()` oder `whichPlatform()`. Diese Objekte geben das verwendete Betriebssystem (OS) bzw. die Plattform aus, sodass das Programm erkennen kann, ob ein "OS-abhängiges" Objekt verwendet werden soll. Die “OS-abhängigen" Objekte sind im Menü durch das Suffix `PC` oder `UNIX` nach dem Namen des Objekts gekennzeichnet.

Benannte Pipes und ActiveX-Funktionen

Die Objekte `To/From Named Pipe (UNIX)` und `ActiveX Automation` auf PCs führen für das jeweilige Betriebssystem dieselben Aufgaben zur Kommunikation mit einem anderen Programm oder einer anderen Anwendung aus.

Rocky Mountain Basic

Die Objekte `Initialize Rocky Mountain Basic (UNIX)` und `To/From Rocky Mountain Basic (UNIX)` sind nur für die Arbeit mit Rocky Mountain Basic unter HP-UX konzipiert.

Die Objekte "Execute Program"

Das Objekt `Execute Program` liegt in zwei verschiedenen Versionen vor: eine für HP-UX, eine für PC. Beide Versionen werden zum Starten anderer Programme oder Anwendungen verwendet.

To/From Stdout, Stderr (UNIX)

Diese Objekte funktionieren zwar auf einem PC, sie sind jedoch mit Dateien implementiert und werden für die allgemeine Programmierung nicht empfohlen. Verwenden Sie diese Objekte nur, wenn Sie ein VEE-Programm von einer HP-UX-Plattform auf den PC portieren.

Schriften und Bildschirmauflösungen

VEE auf einem PC wählt eine Schriftgröße, die bei der verwendeten Bildschirmauflösung ein gutes Schriftbild ergibt. Über das Dialogfenster `File ⇒ Default Preferences` können Sie das Aussehen und die Oberfläche der Programme anpassen. Verwenden Sie die Auswahl `Properties` in den Objektmenüs zum Anpassen einzelner Objekte. Falls erforderlich, wandelt VEE die im Programm gespeicherte Schrift in eine auf dem Zielsystem vorhandene Schrift um. Die besten Ergebnisse erzielen Sie, wenn Sie das Programm auf einem Computer erstellen, dessen Bildschirmauflösung und Schriften denen des Zielsystems möglichst ähnlich sind. Vergewissern Sie sich beim Portieren von Programmen, dass die Bildschirmauflösungen ähnlich sind, um Probleme bei der Portierung zu vermeiden. Weitere Informationen hierzu finden Sie unter `fonts, using` im Index der Online-Hilfe.

Datendateien

ASCII-Datendateien, die mit dem Objekt `To File` erstellt wurden, sollten mit dem Objekt `From File` auf der Plattform HP-UX und der Windows-Plattform gelesen werden können. Binärdateien funktionieren über die Plattformen hinweg nicht, da die Byte-Anordnung bei HP-UX und Windows umgekehrt definiert ist.

Kommunikation mit Rocky Mountain Basic-Programmen

VEE enthält Objekte zur Bereitstellung der Kommunikation zwischen VEE und Rocky Mountain Basic unter HP-UX. Für Workstations der Serie 700 (HP-UX 10.20) können Sie VEE 6.0 verwenden. Für Workstations der Serie 300 und 400 (HP-UX 9.0) müssen Sie HP VEE 3.1 verwenden.

Verwenden des Objekts "Initialize Rocky Mountain Basic"

Das Objekt `Initialize Rocky Mountain Basic` (siehe Abbildung 12-1) enthält ein einzelnes Feld, in dem Sie das auszuführende Rocky Mountain Basic (RMB)-Programm angeben.

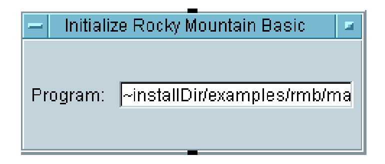


Abbildung 12-1. Das Objekt "Initialize Rocky Mountain Basic"

Geben Sie den vollständigen Pfad und alle erforderlichen Optionen für das Programm an. Das Programm `~installDir/examples/rmb/man34a.bas` könnte beispielsweise in RMB gespeichert worden sein. Das Objekt initiiert den RMB-Prozess und führt das Programm aus. Sie können zur Angabe des Programms auch relative Pfade vom aktuellen Arbeitsverzeichnis aus verwenden. Dieses Objekt stellt keinen Datenpfad zu oder von RMB bereit. Verwenden Sie das Objekt `To/From Rocky Mountain Basic` zur Angabe des Projekts. Sie können mehrere Objekte `Initialize Rocky Mountain Basic` in einem VEE-Programm verwenden.

Hinweis

Es gibt keine direkte Methode zum Beenden eines RMB-Prozesses von einem VEE-Programm aus. Verwenden Sie stattdessen die Anweisung `QUIT` in dem RMB-Programm, wenn es einen bestimmten Datenwert von dem VEE-Programm empfängt. Sie können auch ein Objekt `Execute Program` verwenden, um den RMB-Prozess mit "kill" zu beenden, beispielsweise das Objekt `rmbkill`. Wenn Sie VEE beenden, werden alle noch zugeordneten RMB-Prozesse automatisch beendet.

Verwenden des Objekts "To/From Rocky Mountain Basic"

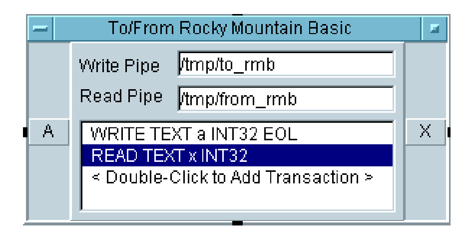


Abbildung 12-2. Das Objekt "To/From Rocky Mountain Basic Object"

Abbildung 12-2 zeigt das Objekt `To/From Rocky Mountain Basic (UNIX)` unter `I/O` ⇒ `Rocky Mountain Basic (UNIX)`. Dieses Objekt ermöglicht die Datenübertragung zu und von RMB-Programmen. Es erstellt und verwendet benannte Pipes für die prozessübergreifende Kommunikation. Der Einfachheit halber implementiert VEE eine Pipe für `READ`-Transaktionen und eine andere Pipe für `WRITE`-Transaktionen.

Im Objekt `To/From Rocky Mountain Basic (UNIX)` in Abbildung 12-2 gibt es zwei Transaktionen: Schreiben einer `Int32`-Ganzzahl und Lesen einer `Int32`-Ganzzahl. Sie können die Standard-Pipes verwenden oder eigene Pipes erstellen, indem Sie neue Pfad- und Dateinamen für `Write Pipe` und `Read Pipe` eingeben. Die Transaktionen werden genau wie andere Transaktionsobjekte in VEE konfiguriert.

Hinweis

Die Felder `Write Pipe` und `Read Pipe` können dem Objekt als Steuer-Pins hinzugefügt werden.

Die folgenden Punkte enthalten weitere Informationen zur Kommunikation mit Rocky Mountain Basic:

Standardnamen für Lese- und Schreib-Pipes

Alle Objekte `To/From Rocky Mountain Basic` enthalten die gleichen Standardnamen für Lese- und Schreib-Pipes. Vergewissern Sie sich daher, dass Sie die richtige Pipe für `READ` bzw. `WRITE` angeben. Vergewissern Sie sich außerdem, dass die Pipes für unterschiedliche Programme eindeutige Namen haben. Achten Sie in dem RMB-Programm darauf, dass die Anweisungen `OUTPUT` und `ENTER` an die richtige Pipe adressiert werden.

Pipes, die automatisch erstellt werden

Wenn vor der Ausführung von `To/From Rocky Mountain Basic` keine Pipes vorhanden sind, werden die folgenden Pipes automatisch erstellt:

`/tmp/to_rmb`

`/tmp/from_rmb`

Das Programm wird jedoch schneller ausgeführt, wenn die Pipes bereits vorhanden sind.

Erstellen zusätzlicher Pipes

Verwenden Sie zum Erstellen zusätzlicher Pipes den Betriebssystembefehl `mknod`.

Öffnen und Schließen von Pipes

Rocky Mountain Basic-Pipes (die nichts anderes sind als benannte Pipes) werden geöffnet, wenn die erste `READ`- oder `WRITE`-Transaktion auf diese Pipe nach einem `PreRun` ausgeführt wird. Alle benannten Pipes werden bei Ausführung von `PostRun` geschlossen. (Weitere Informationen zu `PreRun` und `PostRun` finden Sie im Handbuch *VEE Pro Advanced Techniques*.) Die Transaktionen `EXECUTE CLOSE READ PIPE` und `EXECUTE CLOSE WRITE PIPE` ermöglichen jederzeit das Schließen von Pipes.

Strukturieren von Transaktionen mit Pipes

Wegen des Verhaltens benannter Pipes ist es das einfachste, Transaktionen zum Übertragen bekannter oder einfach unterteilter Datenblöcke zu strukturieren. Wenn Sie beispielsweise Zeichenfolgen übertragen, legen Sie die maximale Länge der zu übertragenden Blöcke fest und füllen Sie kürzere Blöcke mit Leerzeichen auf. Auf diese Weise vermeiden Sie Probleme durch den Versuch, mehr Daten aus einer Pipe zu lesen als tatsächlich vorhanden sind sowie das Problem, dass unerwünschte Daten in der Pipe bleiben.

Verwenden der Transaktion DATA READY

Wenn Sie verhindern wollen, dass eine `READ`-Transaktion blockiert, bis Daten vorliegen, verwenden Sie eine Transaktion `READ IOSTATUS DATA READY` in einem separaten Objekt `To/From Rocky Mountain Basic`. Die Transaktion gibt eine `1` zurück, wenn mindestens 1 Byte zum Lesen vorliegt, und eine `0`, wenn keine Daten vorliegen. Zum Lesen aller an der Lese-Pipe verfügbaren Daten verwenden Sie eine Transaktion `READ ... ARRAY ID TO END:(*)`.

Ausführen in einer Umgebung ohne Platte

Wenn Sie mit einem "Diskless"-System ohne Festplatten- oder Diskettenlaufwerk arbeiten, vergewissern Sie sich, dass Sie `WRITE`- und `READ`-Transaktionen auf eindeutig benannte Pipes ausführen. Andernfalls könnten mehrere Workstations am gleichen Diskless Cluster versuchen, auf dieselbe benannte Pipe zuzugreifen, was zu einem Konflikt führen kann.

Abbildung 12-3 zeigt, wie die Kommunikation mit einem RMB-Programm eingerichtet wird. Dieses Programm befindet sich in `manual34.vee` im Unterverzeichnis `examples/rmb` (nur in der HP-UX-Plattform).

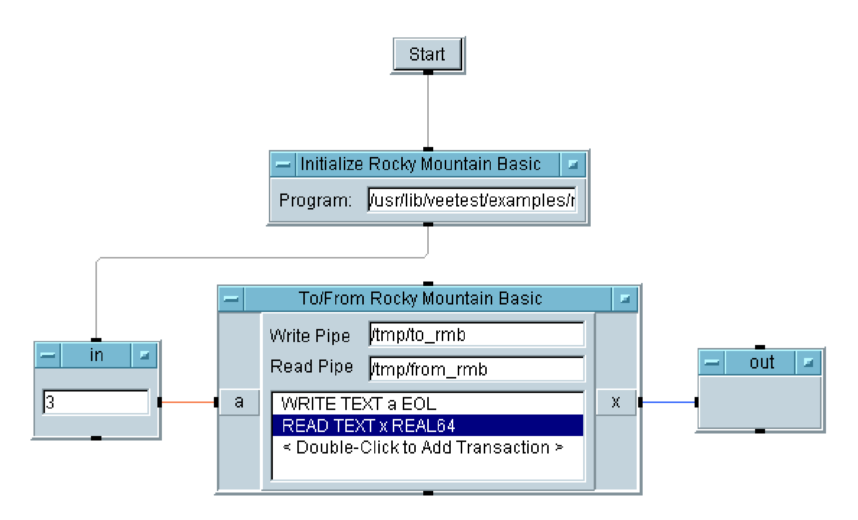


Abbildung 12-3. Kommunikation mit Rocky Mountain Basic

Der Aufrufbare VEE ActiveX-Automatisierungs-Server

In VEE unter Windows 95, Windows 98, Windows NT 4.0 und Windows 2000 können Sie VEE-Objekte in andere kommerziell verfügbare oder eigene Testsysteme integrieren, die in Standardsprachen geschrieben wurden, beispielsweise in MS Visual Basic oder C. Der aufrufbare VEE ActiveX-Automatisierungs-Server kapselt VEE `UserFunctions` zur Integration in Automatisierungsanwendungen ein, beispielsweise in MS Excel und MS Visual Basic. Der aufrufbare VEE ActiveX-Automatisierungs-Server implementiert eine sprachabhängige Automatisierungs-Schnittstelle. Sie kann von jeder Anwendung oder Sprache verwendet werden, die den Aufruf eines Automatisierungs-Servers unterstützt.

Hinweis

VEE kann die ActiveX-Automatisierung und -Steuerelemente zum Steuern anderer Anwendungen von VEE aus verwenden. Der aufrufbare VEE ActiveX-Automatisierungs-Server ermöglicht anderen Anwendungen wie MS Visual Basic die Steuerung von VEE.

Der aufrufbare VEE ActiveX-Automatisierungs-Server kommuniziert mit seinen Clients über Eigenschaften und Methoden. Die Umgebung, von der aus Sie den aufrufbaren VEE ActiveX-Automatisierungs-Server aufrufen (z. B. Visual Basic oder C) legt fest, wie dieser Aufruf erfolgt. Der aufrufbare VEE ActiveX-Automatisierungs-Server bietet mit der Umgebung, von der aus Sie ihn aufrufen, eine umfangreiche Online-`Hilfe`. Weitere Informationen finden Sie in der Online-`Hilfe` in VEE, dem Handbuch *VEE Pro Advanced Techniques* und den zusammen mit VEE ausgelieferten Beispielen.

Hinweis

Der aufrufbare VEE ActiveX-Automatisierungs-Server ersetzt das aufrufbare VEE ActiveX-Steuerelement, das mit VEE Version 5.0 ausgeliefert wurde.

Technologien zur Web-Aktivierung

Sie können mit VEE Daten verbreiten, die sie in Programmen erfasst haben, um Testsysteme zu überwachen oder um Testergebnisse fern zu überprüfen. In diesem Abschnitt werden Technologien zur Web-Aktivierung für Test- und Messanwendungen beschrieben. Außerdem finden Sie hier Informationen dazu, wie VEE diese Technologien unterstützt.

Überblick über Web-Technologien

Dieses Beispiel beschreibt, wie eine organisationsinterne Web-Site (eine "Intranet-Web-Site" erstellt wird, die als Server Referenzdaten zur Verfügung stellt. Dabei wird davon ausgegangen, dass die Benutzer, die auf diese Site zugreifen, einen Browser haben. Das Beispiel verwendet eine Microsoft-Umgebung (Windows 95, Windows 98, Windows NT 4.0 oder Windows 2000), MS Office und MS Internet Explorer. Die Kommunikation erfolgt von dem Instrument an den Server und anschließend an den Client-Browser (siehe Abbildung 12-4).

Hinweis

In diesem Beispiel werden Bildschirmabzüge von einem PC verwendet. VEE ist auch unter HP-UX web-fähig.

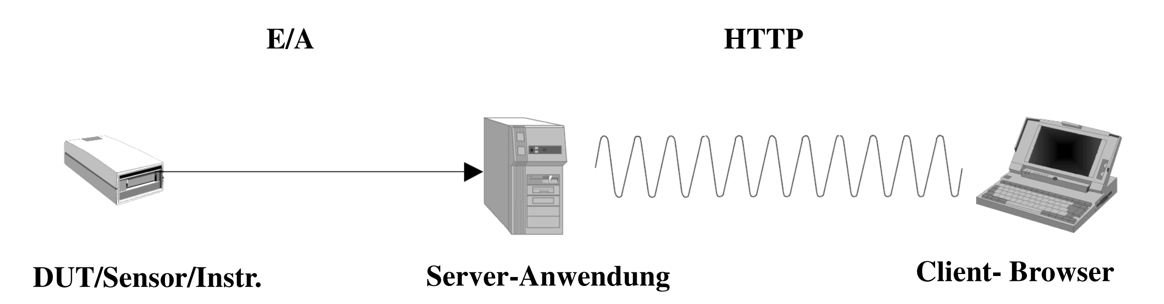


Abbildung 12-4. Modell der Web-Messungsanwendung

In Abbildung 12-4 läuft die Kommunikation wie folgt ab:

- Das *Testgerät ("Device Under Test, DUT") bzw. der Sensor oder das Instrument* sendet Informationen über die Netzwerk-E/A-Schicht an die Server-Anwendung. Die Netzwerk-E/A-Schicht umfasst die Schnittstelle und die Backplane (GPIB, RS232, VIX, MXI, PC Plug-in) und die E/A-Programmierungsschicht wie Treiber, VISA oder SICL.

- Die *Server-Anwendung* ist das VEE-Programm, das die Messdaten generiert.

- Das *HTTP (HyperText Transfer Protocol)* sendet die Informationen an den Client-Browser.

HTTP ist ein Protokoll innerhalb des vom Web verwendeten TCP-Kommunikationsprotokolls. (TCP/IP steht für Transmission Control Protocol/Internet Protocol.) Niedrigere Stufen des TCP/IP-Protokolls umfassen die Transport-, Netzwerk- und die physische Schicht der Kommunikation. Per Definition ist jede TCP/IP-Anwendung eine Client/Server-Anwendung. Der Client-Browser (z. B. Internet Explorer oder Netscape Navigator) kann Informationen anfordern, die von der Server-Anwendung generiert wurden.

Der folgende Universal Resource Locator (URL) kann im Internet Explorer eingegeben werden, um Informationen von einem Agilent -Server abzurufen:

`http://www.agilent.com/find/vee`

- `http` beschreibt den Typ der Ressource, die zum Übertragen der Informationen aufgerufen wurde.

- `www.agilent.com/find/vee` ist der URL für die Ressource. Das von HTTP verwendete Hypertext-Format ist ein Script-Format, das als HyperText Markup Language (HTML) bezeichnet wird. HTML bietet eine Möglichkeit, Dokumente miteinander zu verbinden; ursprünglich war dies die einzige Sprache, mit der Programmierer eine Web-Site erstellen konnten. Ursprünglich nur für Textdaten entwickelt, umfasst HTML heute auch Sound-, Video- und Bilddaten, interaktive Bildschirme, ActiveX-Steuerelemente und Java-Applets.

Wenn der Browser eine Anforderung nach Server-Informationen absetzt, wird er nicht automatisch aktualisiert, es sei denn, der Browser ist entsprechend konzipiert. Außerdem ist keine Interaktion durch den Browser möglich, es sei denn, diese Interaktion ist in der Browser-Seite vorgesehen. Die einfachste Möglichkeit hierzu ist über eine Script-Sprache wie beispielsweise VBScript, JavaScript oder JBscript.

Die Script-Sprache ist eine interpretierte Sprache, die vom Browser unterstützt wird. Die Script-Sprache kann die Einschränkungen von HTML umgehen, um eine Web-Seite mit stärker interaktiven Eigenschaften bereitzustellen. Da sie interpretiert werden, müssen Script-Sprachen in die Web-Seite integriert und vom Browser unterstützt werden. Es handelt sich dabei nicht um unabhängige Programme. Dies wird in Abbildung 12-5 grafisch dargestellt.

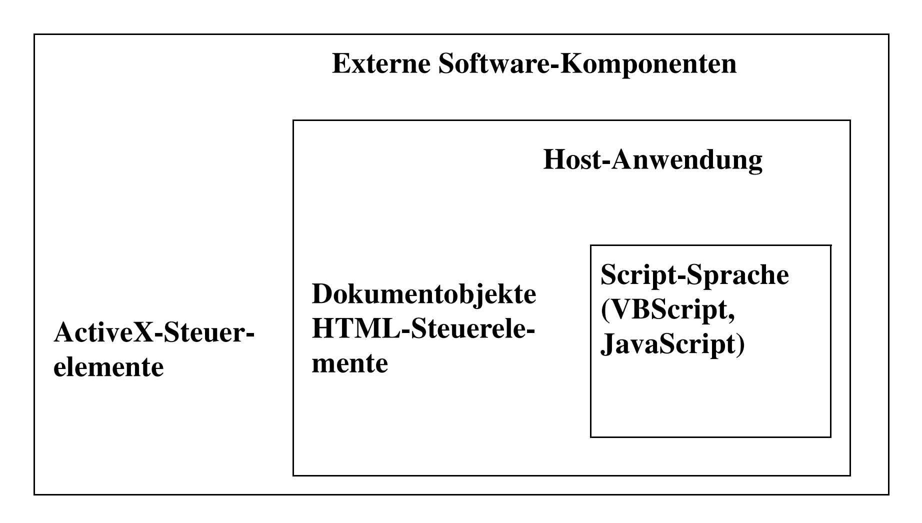


Abbildung 12-5. Ein Host-Modell zur Script-Sprache

VBScript, JavaScript und JScript sind Script-Sprachen. VBScript basiert auf MS Visual Basic. JavaScript wurde von (Java) mitentwickelt. JScript basiert auf der Microsoft Version von JavaScript.

Die Script-Sprachen müssen in einer Host-Anwendung vorhanden sein. Die Host-Anwendung ist normalerweise ein Web-Browser wie Internet Explorer oder Netscape Navigator. Innerhalb des Browsers wird das Aussehen und Verhalten der Web-Seite über HTML gesteuert.

Das Component Object Model (COM) von Microsoft definiert kompilierte Software-Komponenten (ActiveX-Steuerelemente), die speziell konzipierte Funktionen einkapseln. Normalerweise wird ein ActiveX-Steuerelement verwendet, um die Funktionalität einer Benutzerschnittstelle bereitzustellen; es wird auf dem Client-Computer ausgeführt. ActiveX-Steuerelemente sind optimierte Software-Komponenten; sie wurden früher auch als OLE-Steuerelemente bezeichnet.

Web-Überwachung mit Agilent VEE

VEE enthält einen integrierten Web-Server, der über HTTP mit anderen Programmen kommunizieren kann. Sie können fernen Benutzern Zugriff auf VEE-Programme und Informationen auf Ihrem Computer erteilen. In diesem Abschnitt wird beschrieben, wie Sie VEE-Daten für die gemeinsame Nutzung freigeben und wie ein ferner Benutzer auf diese Daten zugreift.

Allgemeine Richtlinien und Tipps

- Das auf Ihrem System ausgeführte VEE-Programm ist das VEE- Programm, auf das der ferne Benutzer zugreift.
- Auf Ihrem System muss VEE aktiv sein. Der ferne Benutzer braucht VEE nicht installiert zu haben, um auf ein VEE-Programm auf Ihrem System zugreifen zu können.
- Wenn der ferne Benutzer eine Anforderung von seinem Netzwerk-Browser an VEE sendet, erstellt VEE ein Bild zur Anzeige im Browser-Fenster des fernen Benutzers. Dieses Bild ist eine "Momentaufnahme" des VEE-Programms; es kann nicht geändert werden.
- Durch das Auswählen von Optionen in der VEE Web-Server-Homepage oder durch Eingeben von Befehlszeilenoptionen im Browser-URL kann ein Benutzer verschiedene Teile Ihres VEE-Programms anzeigen, das VEE-Programm die Browser-Anzeige in regelmäßigen Abständen aktualisieren lassen (zur Überwachung des Programmfortschritts) und Informationen zu Fehlermeldungen anzeigen.

Bereitstellen von Agilent VEE-Daten an einen fernen Benutzer

Zum Einrichten des VEE Web-Servers, sodass ferne Benutzer auf Daten auf ihrem System zugreifen können, führen Sie die folgenden allgemeinen Schritte aus:

1. Überprüfen Sie, ob Ihr System mit dem Netzwerk verbunden ist.

2. Stellen Sie dem fernen Benutzer Informationen über den URL zur Verfügung; diesen URL muss der Benutzer in seinem Browser eingeben, um auf Ihr System zuzugreifen (dies wird weiter unten ausführlich beschrieben).

3. Starten Sie VEE und öffnen Sie das Programm, das der ferne Benutzer aufrufen soll bzw. erstellen Sie alle Dateien, die Sie dem fernen Benutzer zur Verfügung stellen wollen.

4. Aktivieren Sie den Web-Server durch Auswählen der Einstellungen im Dialogfenster `File` ⇒ `Default Preferences` ⇒ `Web Server`; diese Einstellungen werden weiter unten ausführlich beschrieben.

5. Der ferne Benutzer muss jetzt einen Web-Browser wie Internet Explorer oder Netscape aufrufen.

Dialogfenster "Web Server"

Wenn Sie `File` ⇒ `Default Preferences` ⇒ `Web Server` auswählen, wird das Dialogfenster `Web Server` angezeigt (siehe Abbildung 12-6).

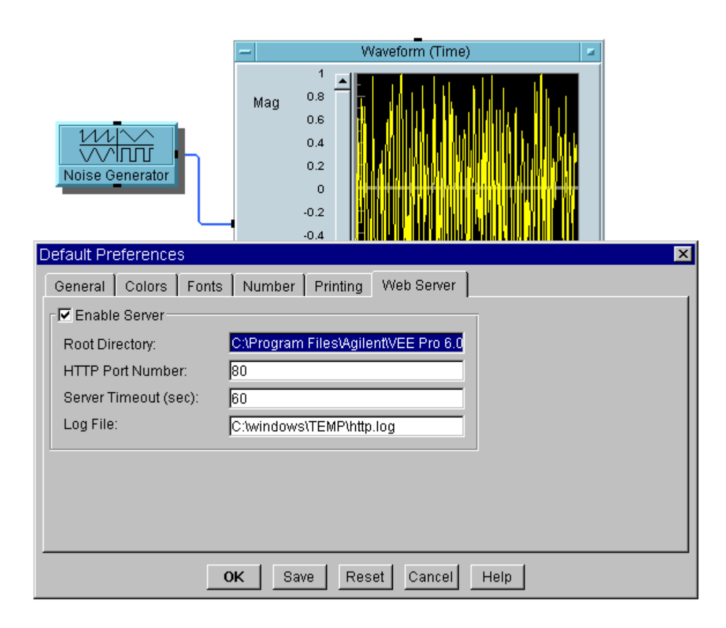


Abbildung 12-6. Das Dialogfenster "Default Preferences Web Server"

Die Felder im Dialogfenster `Web Server` sind:

Enable Server (Server aktivieren)

Wenn diese Option ausgewählt ist, schaltet `Enable Server` den integrierten VEE Web-Server ein. Der Web-Server ermöglicht einem fernen Benutzer das Anzeigen, Überwachen und Beheben von Fehlern mit VEE-Programmen auf Ihrem System durch Verwenden eines Web-Browser zum Anzeigen Ihres VEE-Programms.

Der VEE Web-Server verwendet das Standard-HTTP-Protokoll. Standardmäßig ist `Enable Server` auf `OFF` (nicht markiert) eingestellt. Ihr System muss mit einem Netzwerk verbunden sein, VEE muss aktiv sein, und der ferne Benutzer muss einen Web-Browser aufgerufen haben, damit er auf den VEE Web-Server zugreifen kann.

Root Directory (Root-Verzeichnis)

Gibt die Position der Dateien an, die von dem fernen Benutzer aufgerufen werden können. Der Standardwert lautet ~*installDir*`/www`. Für Windows 95, Windows 98, Windows NT 4.0 und Windows 2000 lautet diese Datei, sofern VEE Pro 6.0 mit den Standardwerten installiert ist, wie folgt:

`C:\Program Files\Agilent\VEE Pro 6.0\www`.

- Legen Sie *keine* privaten Dateien in diesem Verzeichnis oder seinen Unterverzeichnissen ab, da sie von allen fernen Benutzern, die mit einem Web-Browser Zugriff auf Ihr System haben, eingesehen werden können.
- VEE installiert bei der Installation von VEE die Datei `index.html` im `Root Directory` des Standard-Web-Servers. Standardmäßig befindet sich die Datei im Verzeichnis `C:\Program Files\Agilent\VEE Pro 6.0\www\index.html`. Dies entspricht der Standard-Homepage des VEE-Web-Servers; sie kann beim Zugriff auf Ihr System durch einen fernen Benutzer angezeigt werden. (Sie können die Homepage an Ihre Anforderungen anpassen.)
- Wenn Sie das Root-Verzeichnis für Web-Dateien ändern, verschieben Sie die Datei `index.html` in das neue Verzeichnis und ändern Sie ihren Namen nicht.

HTTP Port Number (HTTP-Anschlussnummer)

Gibt die Anschlussnummer für den VEE Web-Server an. Die Standardanschlussnummer lautet 80. Sie brauchen die Anschlussnummer nur zu ändern, wenn ein anderer Web-Server, beispielsweise ein anderes Exemplar von VEE, auf dem gleichen System aktiv ist, oder wenn Sie den Fernzugriff auf Ihren VEE Web-Server beschränken wollen.

- Durch die Angabe eines HTTP-Anschlusses zwischen 0 und 65535 können Sie begrenzen, wer Zugriff auf Ihr System hat und Daten anzeigen darf.
- Unter HP-UX sind nur Anschlussnummern größer als 1024 zulässig. Die Standard-Anschlussnummer lautet 8080.
- Wenn Sie eine andere Anschlussnummer eingeben, muss der ferne Benutzer, der das VEE-Programm anzeigen will, die gleiche Anschlussnummer in seinem Browser angeben. Wenn Sie das Feld beispielsweise auf 85 setzen, muss der ferne Benutzer den URL als `http://`*hostname*`:85` eingeben. (Weitere Informationen zur Definition des *hostname* finden Sie weiter hinten in diesem Kapitel.)

Server Timeout (Server-Zeitüberschreitung)

Gibt die maximale Zeit an, die der VEE Web-Server darauf wartet, dass VEE Befehle verarbeitet. Standardmäßig ist dieses Feld auf 60 Sekunden gesetzt. Wenn das VEE-Programm zur Verarbeitung eines Befehls länger als die angegebene Zeit braucht, sendet VEE eine Timeout-Signal an den Web-Server.

Log File (Protokolldatei)

Gibt die Protokolldatei an, in der alle vom Web-Server verarbeiteten ankommenden HTTP-Anforderungen aufgezeichnet werden. Die Standardposition für diese Datei auf einem PC ist `C:\windows\TEMP\http.log`. Wenn die Protokolldatei nicht bereits vorhanden ist, wird sie erstellt, wenn Sie im Dialogfenster `Default Preferences` ⇒ `Web Server` Änderungen vornehmen.

Save (Speichern) Speichert die als permanente Auswahl angezeigten Attribute. Dadurch werden die Einstellungen in der Datei `vee.rc` unter Windows bzw. `.veerc` unter HP-UX gespeichert.

Diese Werte stellen die Standardwerte für zukünftige VEE Web-Sessions dar, bis Sie die erneut ändern. Wenn Sie die angegebenen Werte *nur* für die aktuelle VEE-Session verwenden wollen, klicken Sie `OK` an.

Wie ein ferner Benutzer auf Agilent VEE auf Ihrem System zugreift

Für den Zugriff auf VEE-Dateien auf Ihrem System von einem anderen Standort aus muss der ferne Benutzer die folgenden allgemeinen Schritte durchführen:

1. Das System des fernen Benutzers muss mit einem Netzwerk verbunden sein.
2. Der ferne Benutzer muss einen Netzwerk-Browser wie beispielsweise Netscape oder Internet Explorer aktivieren.
3. Der ferne Benutzer muss eine URL-Adresse für Ihr System eingeben. (Sie müssen diese Adresse dem fernen Benutzer mitteilen; dies wird weiter unten ausführlich erläutert.)

Hinweis Der ferne Benutzer braucht VEE nicht zu aktivieren und muss VEE auch nicht installiert haben, um auf VEE-Programme auf Ihrem System zugreifen zu können.

Der ferne Benutzer stellt eine Verbindung zum Netzwerk her, startet den Browser und gibt die URL-Adresse ein. Sie teilen dem fernen Benutzer die URL-Adresse entsprechend den folgenden Angaben mit. Der ferne Benutzer kann diese Adresse in einem der folgenden Formate eingeben:

`http://hostname {:port}`

Zeigt die Homepage Ihres VEE Web-Servers an, in der der ferne Benutzer Optionen zur Anzeige Ihres VEE-Programms auswählen kann.

`http://hostname{:port}{/Befehl}{?Parameter}`

Gibt eine Ansicht oder `UserFunction` an, die der ferne Benutzer in dem VEE-Program anzeigen will. `http://`*hostname*`/ViewMainDetail` zeigt beispielsweise die Detailansicht Ihres VEE-Hauptprogramms an.

`http://hostname{:port}{/Datei}`

Gibt eine von Ihnen gespeicherte Datei an, z. B, eine `*.jpeg-` oder

`*.html`-Datei, die dem fernen Benutzer angezeigt werden soll.

Hinweis Die Felder in geschweiften Klammern { } sind optionale Felder.

Die Felder in den URL-Adressen haben folgende Bedeutung:

Hostname (Erforderlich) Kennzeichnet Ihr System, auf dem VEE aktiv ist, im Format <*computer_name*>`.domain.com`.

Sie können festlegen, dass der Benutzer lediglich den <*hostname*> in einer URL-Adresse anzugeben braucht, z. B. `http://`<*hostname*>. Dieser Befehl öffnet die Homepage Ihres VEE-Web-Servers `index.html` und zeigt sie dem fernen Benutzer an. Der ferne Benutzer kann in der Homepage Ihres VEE Web-Servers Optionen auswählen zur Anzeige, zur Überwachung oder zur Fehlerbehebung Ihres VEE-Programms.

Port (Anschluss) (Optional) Kennzeichnet die Anschlussnummer des Web-Servers, wenn nicht der Standardwert 80 verwendet werden soll. Geben Sie diesen Wert nur an, wenn Sie unter `File` ⇒ `Default Preferences` ⇒ `Web Server` eine andere Nummer als 80 eingegeben haben.

Wenn der ferne Benutzer beispielsweise auf VEE auf Ihrem System zugreift und die Anschlussnummer auf Ihrem System unter `File` ⇒ `Default Preference` ⇒ `Web Server` auf `85` gesetzt ist, muss der ferne Benutzer `http://`<*hostname*>`:85` eingeben.

File (Datei)	(Optional) Kennzeichnet ein Verzeichnis und/oder eine Datei relativ zu dem Root-Verzeichnis, das von dem Browser geöffnet wird. Sie geben nur eine Datei an, wenn Sie eine Datei wie beispielsweise eine `*.jpeg`- oder `*.html`-Datei gespeichert haben, die dem Benutzer angezeigt werden soll. Damit die fernen Benutzer eine Datei von Ihrem System anzeigen können, müssen Sie das Verzeichnis als `Root-Verzeichnis` im Dialogfenster `Web Server` unter `Default Preferences` in VEE angeben.
Commands and Parameters (Befehle und Parameter)	(Optional) Gibt einen Befehl oder einen für einen Befehl erforderlichen Parameter an, der vom VEE Web-Server unterstützt wird. Befehle und Parameter geben einem fernen Benutzer die Möglichkeit, ein fernes VEE-Programm über den Browser zu überwachen und Fehler zu beheben. Nachfolgend finden Sie eine Liste der Befehle und Parameter.

Folgende Befehle und Parameter können in URLs verwendet werden, wenn ein ferner Benutzer auf ein VEE-Programm auf Ihrem System zugreift.

Anzeigen des gesamten VEE-Fensters	`ViewVEE` Beispiel: `http://`*hostname*`/ViewVEE`

Fensteransicht des VEE-Hauptprogramms	`ViewMainPanel`
	
	`Beispiel:`
	`http://`*hostname*`/ViewMainPanel`

Detailansicht des VEE-Hauptprogramms	`ViewMainDetail`
	
	`Beispiel:`
	`http://`*hostname*`/ViewMainDetail`

VEE -Ausführungs-fenster zur Ausführungszeit	`ViewExecWindow`
	
	`Beispiel:`
	`http://`*hostname*`/ViewExecWindow`

VEE UserFunction in der Fensteransicht	`http://`*hostname*`/ViewPanel?` <*UserFunctionName*> Wenn die `UserFunction` beispielsweise `AddNoise` lautet, gibt der ferne Benutzer ein: `http://`*hostname*`/ViewPanel?` `AddNoise`
Detailansicht einer UserFunction	`http://`*hostname*`/ViewDetail?` `UserFunctionName` Wenn die `UserFunction` beispielsweise `AddNoise` lautet, gibt der ferne Benutzer ein: `http://`*hostname*`/ViewPanel?` `AddNoise`
Fehlerfenster zum aktuellen Programm	`http://`*hostname*`/ViewError`
Anzeigen einer Liste der verfügbaren Befehls-URLs.	`ViewHelp`

Anzeigen der Homepage des Agilent VEE-Web-Servers

Bei der Installation von VEE wird die Standarddatei `index.html` im Verzeichnis `www` erstellt. Diese Datei enthält die Homepage des VEE-Web-Servers. Sie können fernen Benutzern die Möglichkeit geben, auf dieser Seite Optionen zur Anzeige Ihres VEE-Programms anzuklicken. Die Standard-Homepage des VEE-Web-Servers wird in Abbildung 12-7 dargestellt. Sie können diese Seite auch beispielsweise in MS Word bearbeiten, um sie an Ihre Anforderungen anzupassen.



Abbildung 12-7. Die Standardseite Index.html

Hinweis

Zum Anzeigen dieses Menüs auf Ihrem eigenen System können Sie auf Ihr System als `localhost` verweisen. Wenn Sie beispielsweise VEE ausführen, den Netzwerk-Browser starten und `http://localhost`eingeben, zeigt der Browser die Homepage des VEE-Web-Servers in Abbildung 12-7 an. Dies ist eine einfache Möglichkeit für Sie, zu prüfen, welche Anzeigen der ferne Benutzer mit den verschiedenen Befehlen erhält.

Wenn der ferne Benutzer das in Abbildung 12-7 dargestellte Menü anzeigt, kann er auf verschiedene Bereiche eines VEE-Programms zugreifen, indem er in dem Menü Optionen auswählt. Der Benutzer kann beispielsweise `Main Detail` anklicken, um das VEE-Hauptprogramm in der Detailansicht anzuzeigen. Durch die Auswahl im Menü werden die gleichen Informationen angezeigt, die der ferne Benutzer sieht, wenn er den Befehl `http://`*hostname*`/ViewMainDetail` im Netzwerk-Browser eingibt.

Übung 12-1: Übungs-Session mit Agilent VEE Web-Browser

Diese Übung simuliert eine Web-Session, in der Sie das Programm `Solitaire.vee` auf Ihrem System für einen fernen Benutzer bereitstellen. In diesem Fall tritt ein Fehler im Programm auf, und der ferne Benutzer versucht mit Ihnen zusammen, den Fehler zu beheben.

1. Starten Sie VEE. Wählen Sie das Dialogfenster `File` ⇒ `Default Preferences` ⇒ `Web Server` aus und klicken Sie `Enable Server` an. Verwenden Sie die Standardeinstellungen. Öffnen Sie das Programm `Solitaire.vee`, das der ferne Benutzer anzeigen soll. Starten Sie Ihren Netzwerk-Browser.

2. Nehmen Sie Kontakt mit dem fernen Benutzer auf und teilen Sie ihm mit, dass er Ihr System über das Web mit der Eingabe `http://Server5` erreichen kann. (Statt `http://Server5` würden Sie den Namen Ihres Computers verwenden.)

3. Wenn der ferne Benutzer den URL `http://Server5` eingibt, sieht er die Homepage Ihres VEE-Web-Servers in seinem Browser. (In Abbildung 12-7 auf Seite 484 ist die Anzeige dargestellt.)

4. Der ferne Benutzer beschließt, zunächst das gesamte VEE-Programm anzuzeigen. In der Homepage Ihres VEE-Web-Servers klickt er `Main Detail` ⇒ `View` an. Der Browser zeigt die in Abbildung 12-8 dargestellte Anzeige an.

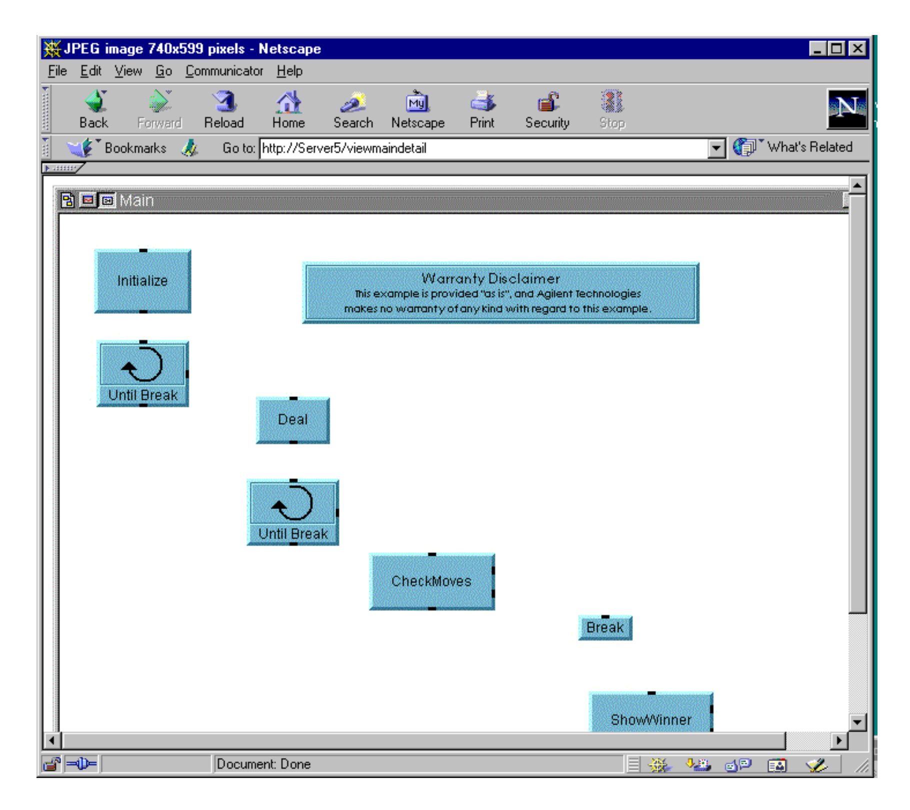


Abbildung 12-8. Anzeigen des Hauptprogramms Solitaire.vee im Browser

Abbildung 12-8 zeigt das Hauptprogramm in VEE an.

Hinweis

Für diese Übung enthält das Programm `Solitaire.vee` einen Fehler, der *nicht* im VEE-Programmbeispiel enthalten ist. Falls Sie das Programm anzeigen wollen: es befindet sich in `Help` ⇒ `Open Example...` ⇒ `Games` ⇒ `Solitaire.vee`.

5. Der ferne Benutzer klickt `Zurück` im Browser an, um die Homepage des VEE-Web-Servers anzuzeigen, und wählt anschließend `Last Error Message` aus. Der Browser zeigt die in Abbildung 12-9 dargestellte Fehlermeldung an.

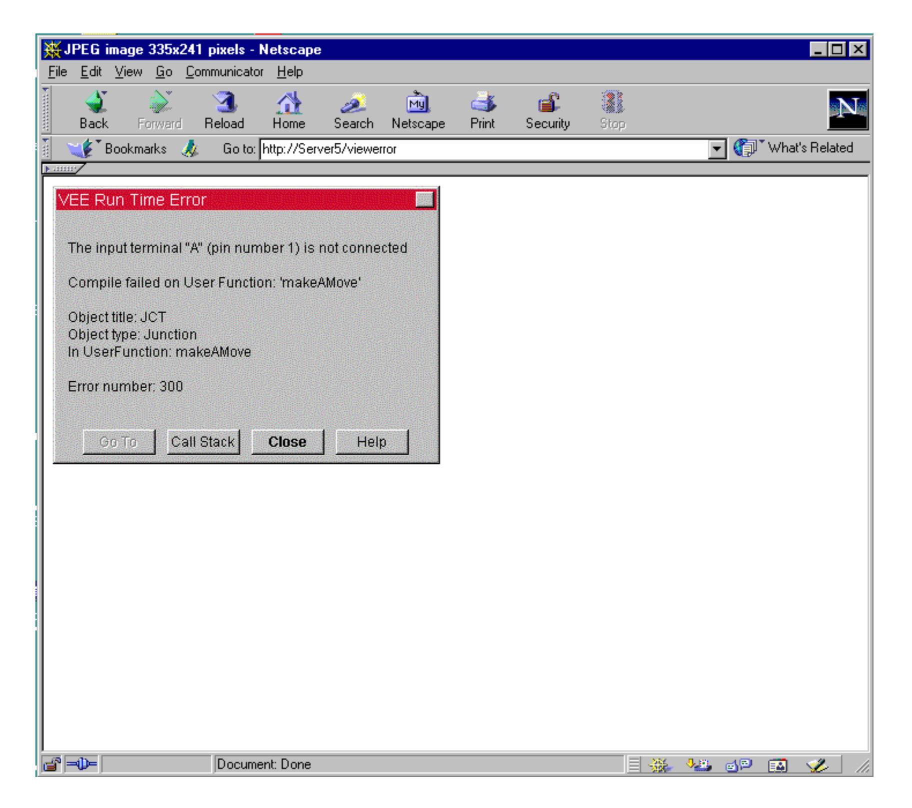


Abbildung 12-9. Anzeigen einer VEE-Fehlermeldung mit dem Browser

Beachten Sie, dass die VEE- Fehlermeldung die `UserFunction makeAMove` angibt.

6. Der ferne Benutzer kehrt wieder zurück zur Homepage des VEE-Web-Servers, klickt `Detail View of UserFunction` an und gibt den den `UserFunction`-Namen `makeAMove` an. Der Browser zeigt die `UserFunction makeAMove` an (siehe Abbildung 12-10).

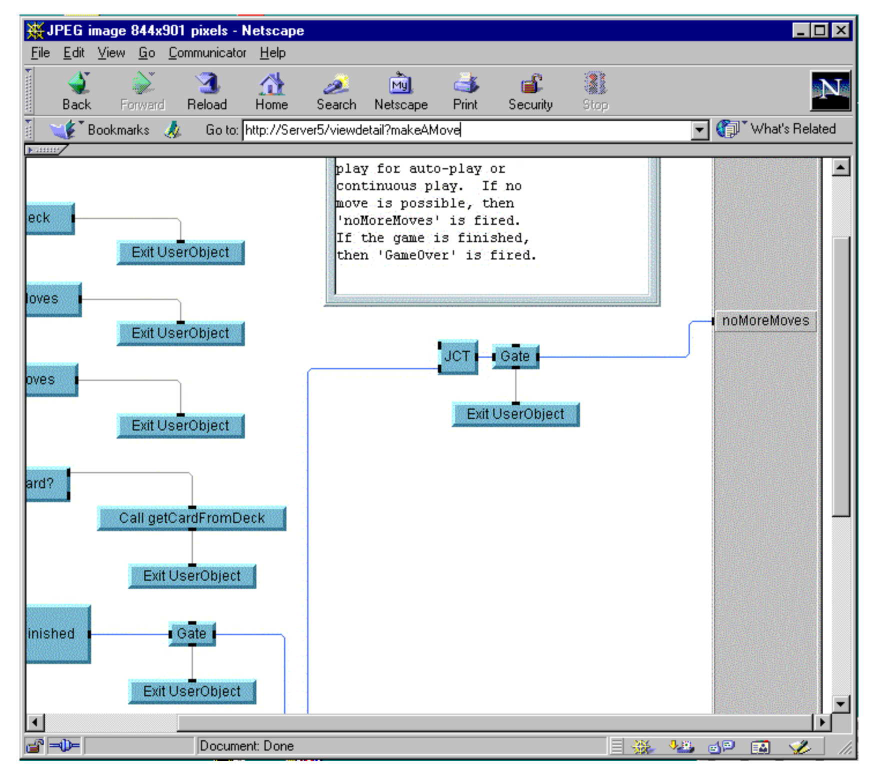


Abbildung 12-10. Die Detailansicht einer UserFunction im Browser

Der ferne Benutzer kann den Fehler im VEE-Programm sehen. Ein Eingangs-Pin am Objekt `JCT` ist nicht angeschlossen (siehe Abbildung 12-10). Der ferne Benutzer könnte Ihnen jetzt bei der Behebung des Fehlers in `Solitaire.vee` helfen und das Problem lösen. Über einen ähnlichen Prozess der Zusammenarbeit über das Web können Sie mit fernen Benutzern gemeinsam Programme entwickeln.

Begrenzen des Zugriffs auf Programme über das Web

Wenn Sie ein VEE-Programm über das Web verfügbar machen, wollen Sie eventuell manche Teile des Programms gegen die Blicke der fernen Benutzer schützen. Wenn die fernen Benutzer derzeit die URL-Adresse zu Ihrem System kennen, wollen Sie sicherstellen, dass nur bestimmte ferne Benutzer auf bestimmte Programme oder Dateien im Web-Verzeichnis zugreifen können.

Wenn Sie verhindern wollen, dass ferne Benutzer Teile eines VEE-Programms über das Web anzeigen, können Sie das Programm auf drei verschiedene Arten schützen:

1. Ändern der Anschlussnummer im Ordner `Default Preferences` ⇒ `Web Server`, sodass nur Benutzer mit entsprechender Berechtigung das Programm anzeigen können.

-ODER-

Erstellen einer gesicherten `Laufzeitversion` des VEE-Programms. Auf diese Weise wird sichergestellt, dass der Programmcode nicht angezeigt werden kann. Weitere Informationen hierzu finden Sie unter “Sichern eines Programms (Erstellen einer Laufzeitversion)” auf Seite 400.

-ODER-

Erstellen einer HTML-Datei mit dem Namen des Befehls, den Sie inaktivieren wollen, und Speichern dieses Befehls im VEE- Verzeichnis `www`. Der Browser greift immer auf jede `*.html`-Datei zu, bevor er zu VEE übergeht. Auf diese Weise können Sie Anforderungen von fernen Benutzern abfangen und eine HTML-Seite mit den entsprechenden Warnungen oder Kommentaren anzeigen.

Sie können beispielsweise verhindern, dass ferne Benutzer die Detailansicht eines VEE-Programms anzeigen. Sie können eine Datei in einem Programm wie beispielsweise MS Word erstellen und als `ViewMainDetail.html` im Verzeichnis `www` speichern. In der Datei legen Sie die Meldung ab, die der Benutzer sehen soll.

Wenn der ferne Benutzer `Main Detail` in der Homepage des VEE-Web-Servers auswählt oder eine URL-Adresse mit der Option `ViewMainDetail` auswählt, zeigt der Browser *nicht* die Detailansicht des VEE-Hauptprogramms an. Stattdessen greift der Browser auf die Datei `ViewMainDetail.html` im Verzeichnis `www` zu und zeigt die von Ihnen erstellte Datei an. Abbildung 12-11 zeigt ein Beispiel für eine Meldung, die Sie einem fernen Benutzer anzeigen könnten.

Hinweis

Vergewissern Sie sich, dass der Dateiname der Dateiname eines VEE-Web-Befehls ist und sich in dem `Root Directory` befindet, das unter `Web Server` angegeben wurde.

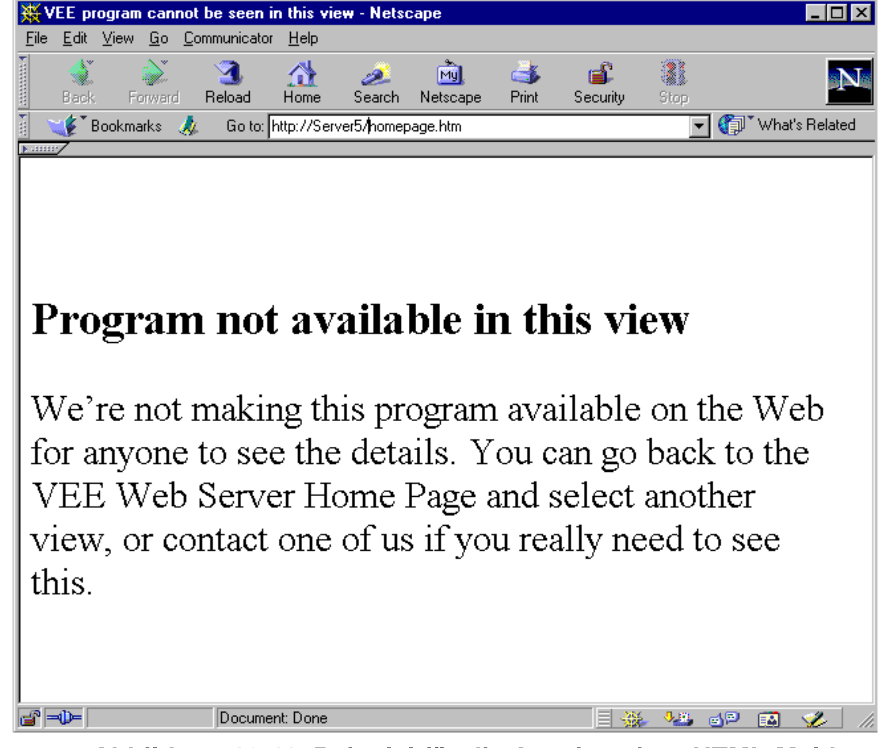


Abbildung 12-11. Beispiel für die Anzeige einer HTML-Meldung statt eines VEE-Programms

Sie können auch eine `*.html`-Datei für andere Zwecke erstellen, etwa um einen Kennwortschutz auf ein VEE-Programm zu legen, so dass nur Benutzer mit dem entsprechenden Kennwort das Programm anzeigen können. Abbildung 12-12 zeigt ein Beispiel für den Kennwortschutz.

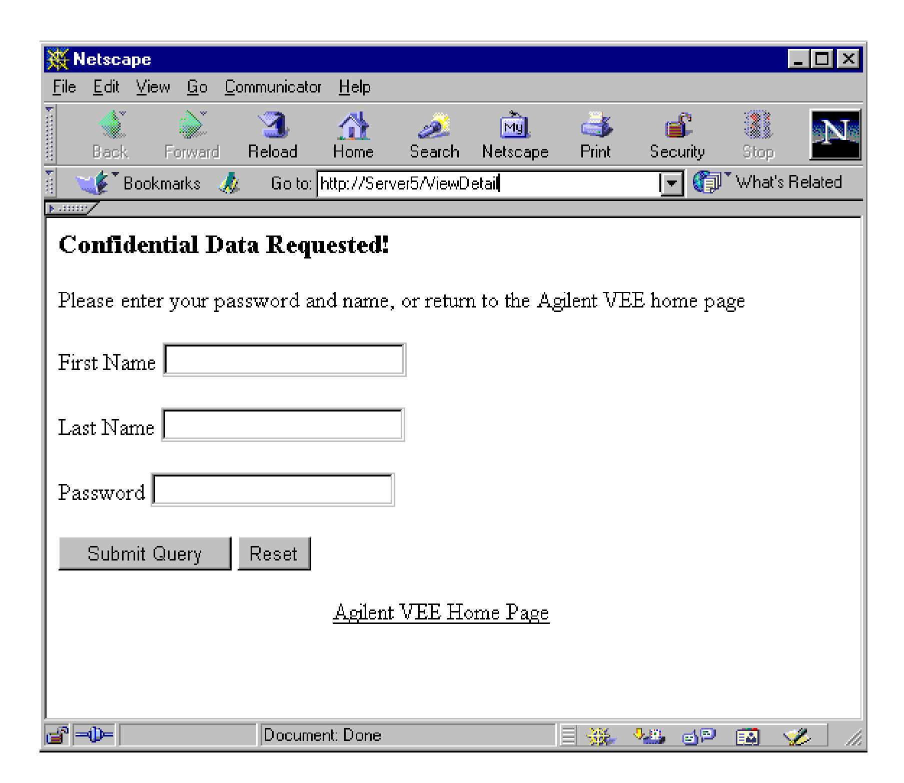


Abbildung 12-12. Ein Beispiel eines Kennwortfensters

Checkliste für das Kapitel

Sie sollten jetzt in der Lage sein, die folgenden Aufgaben auszuführen.

- Erläutern der wichtigsten Unterschiede zwischen PC- und HP-UX-Plattformen und den Faktoren, die beim Portieren von Programmen zu beachten sind.
- Erläutern des Aufrufs und der Kommunikation mit einem Rocky Mountain Basic-Programm.
- Erläutern der Verwendung des aufrufbaren VEE ActiveX-Automatisierungs-Servers und wann er eingesetzt werden sollte.
- Erläutern der Integration der VEE-Funktionalität in anderen Anwendungen oder Programmen.
- Erläutern der wichtigsten Konzepte bei der Verwendung des Web bei der Überwachung von VEE-Programmen.

A

Zusätzliche Übungen

Zusätzliche Übungen

Mit den folgenden Übungen haben Sie die Möglichkeit, weitere praktische Erfahrungen mit den in diesem Handbuch gelernten Konzepten von VEE zu sammeln. Die Übungen sind in vier Kategorien unterteilt.

Sie verwenden diesen Anhang, indem Sie eine Lösung entwickeln und mit den aufgelisteten Antworten vergleichen. Es gibt viele Möglichkeiten, eine vorgegebene Aufgabe zu programmieren. Ihre Lösung ist gültig, wenn Sie die gestellte Aufgabe erfüllt. Programme, die leicht zu bedienen sind und schnell ausgeführt werden, bilden natürlich eine bessere Lösung. Jede Lösung umfasst eine kurze Beschreibung der wichtigsten Punkte.

Allgemeine Programmiertechniken

Der Äpfelsammler

Sie wollen wissen, wie viele Äpfel man braucht, um einen Zehn-Pfund-Korb zu füllen. Erstellen Sie ein VEE-Programm, das zählt, wie viele Äpfel benötigt werden, bis der Korb gefüllt ist. Jeder Apfel wiegt zwischen 0 und 1 Pfund.

Vorschläge

Dieses Programm kann mit 10 oder weniger Objekten erstellt werden. Wählen Sie Objekte aus der folgenden Liste aus:

Start
Until Break
random()-Funktion
Accumulator
Break
Real64
Conditional (A>=B)
Stop
Counter
If/Then/Else
Alphanumeric

Hinweis

Die VEE -Programme für viele der praktischen Übungen und Programmbeispiele in diesem Handbuch sind in VEE unter `Help` ⇒ `Open Example...` ⇒ `Manual` ⇒ `UsersGuide` enthalten.

Lösung 1—der Äpfelsammler

Abbildung A-1 zeigt eine Lösung zur Übung Äpfelzähler.

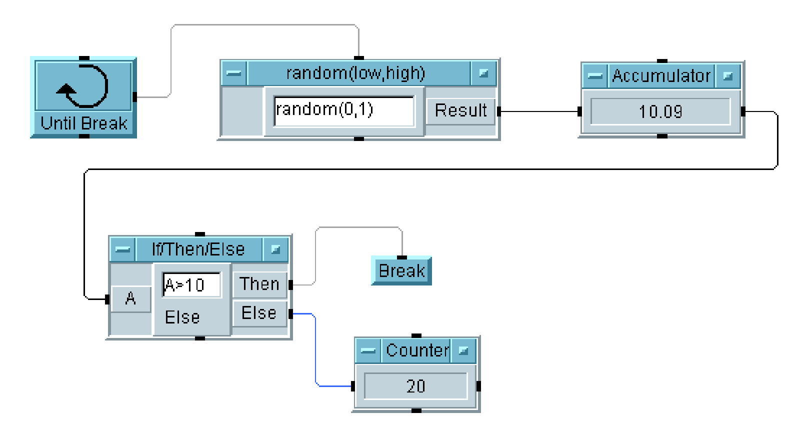


Abbildung A-1. Äpfelzähler, Lösung 1

Wichtige Punkte

- **Optimale Lösungen:** Verwenden Sie zur Optimierung der Leistung von Programmen möglichst wenige Objekte. Diese Lösung verwendet sechs Objekte. Das Programm kann auch mit 10 Objekten implementiert werden, wie in Abbildung A-2 gezeigt.

- **Objekte "Until Break" und "Break":** Verwenden Sie diese Objekte für Schleifen, die das Testen einer Bedingung erfordern. In diesem Beispiel sollte die Schleife stoppen, wenn das Gesamtgewicht der Äpfel größer als zehn Pfund ist.

- **Accumulator:** Verwenden Sie das Objekt `Accumulator` zum Erfassen einer laufenden Summe.

- **Counter:** Verwenden Sie das Objekt `Counter` zum Erfassen einer laufenden Anzahl. In diesem Beispiel wird mit dem Objekt `Counter` die Gesamtzahl der Äpfel im Korb aufgezeichnet. Wenn das Gesamtgewicht größer als 10 ist, löst nur der `Then`-Pin am Objekt `If/Then/Else` aus und liefert die richtige Antwort im Objekt `Counter`.

Lösung 2—der Äpfelsammler

Abbildung A-2 bietet eine andere Lösung mit mehr Objekten.

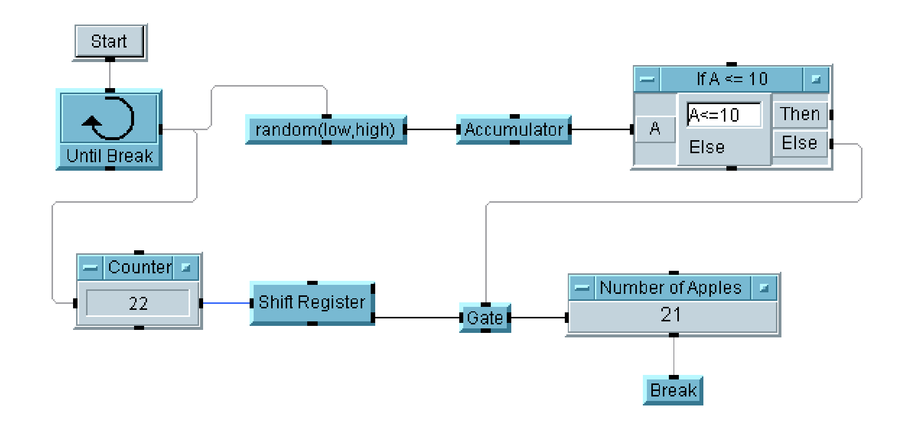


Abbildung A-2. Äpfelsammler, Lösung 2

Wichtige Punkte

- **Start:** Die Verwendung eines `Start`-Objekts für dieses Programm ist redundant, da Sie die Schaltfläche **Ausführen**in der Hauptmenüleiste verwenden können. `Start` wird am besten verwendet, wenn Sie zwei Programme auf dem Bildschirm haben und diese unabhängig voneinander ausführen wollen. Sie können auch ein Programm mit einer Rückmeldungsschleife haben und definieren, wo die Ausführung beginnen soll.

- **Shift Register:** Verwenden Sie ein `Shift Register` für den Zugriff auf die verschiedenen Werte der Ausgabe. In Lösung 2 zeichnet der `Counter` mit jedem eingesammelten Apfel die aktuelle Anzahl auf, bevor der Apfel gewogen wird. Wenn das Gesamtgewicht 10 überschreitet, muss daher die Anzahl wieder um 1 reduziert werden.

- **Gate:** Mit dem Objekt `Gate` wird die Ausgabe gespeichert, bis eine andere Aktion auftritt und den Sequenz-Pin aktiviert. Wenn die Bedingung `A<=10` nicht mehr erfüllt ist, aktiviert der `Else`-Pin am Objekt `If/Then/Else` das Gate-Objekt.

Testen von Zahlen

Testen von Zahlen, Schritt 1

Erstellen Sie ein Programm, mit dem ein Benutzer eine Zahl zwischen 0 und 100 eingeben kann. Wenn die Zahl größer oder gleich 50 ist, soll sie angezeigt werden. Ist die Zahl niedriger als 50, soll ein Fenster mit der Meldung “Bitte eine Zahl zwischen 50 und 100 eingeben” eingeblendet werden.

Vorschläge

Dieses Programm kann mit 5 oder weniger Objekten erstellt werden. Wählen Sie Objekte aus der folgenden Liste aus:

Start
Int32
Slider
Real64
If/Then/Else
Formula
Gate
Text
Junction
Alphanumeric
Message Box

Lösung—Testen von Zahlen, Schritt 1

Abbildung A-3 zeigt eine Lösung zur Übung "Testen von Zahlen" mit fünf Objekten.

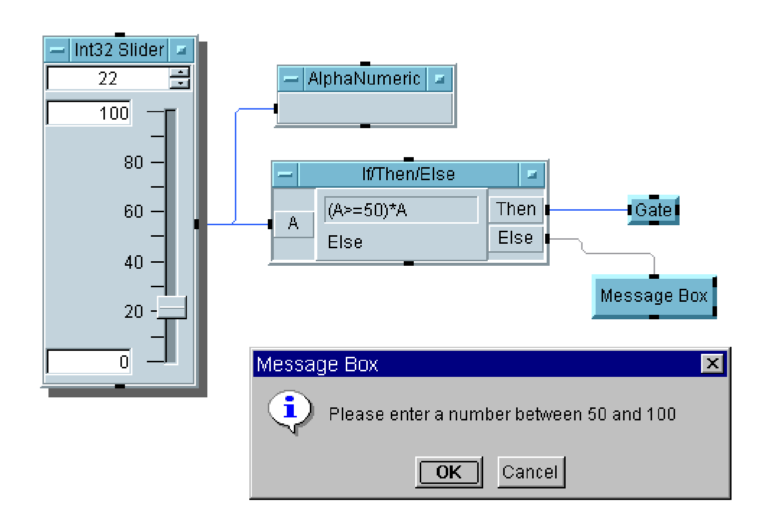


Abbildung A-3. Testen von Zahlen (mit Einblendfenster)

Testen von Zahlen, Schritt 2

Nachdem das Modell mit fünf Objekten funktioniert (die Meldung wird im Einblendfenster angezeigt), versuchen Sie, die Aufgabe mit vier Objekten (ohne das Objekt `Gate`) zu programmieren.

Lösung—Testen von Zahlen, Schritt 2

Abbildung A-4 zeigt die Lösung zur Übung "Testen von Zahlen" mit vier Objekten.

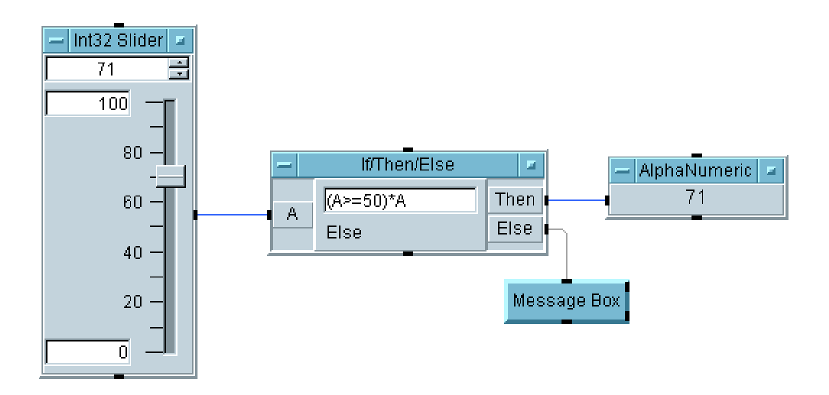


Abbildung A-4. Testen von Zahlen, Schritt 2

Wichtige Punkte

- **Auto Execute:** Alle Eingangsobjekte wie beispielsweise der `Int32 Slider` enthalten eine Auswahl `Auto Execute` im Feld `Properties`. Bei dieser Auswahl arbeitet das Objekt immer, wenn sich sein Wert ändert, ohne dass Sie die Schaltfläche `Start` oder **Ausführen** drücken müssen.

- **Gates eliminieren:** Der Ausdruck `(A>=50)*A` im Objekt `If/Then/Else` wird zu `1*A` ausgewertet, wenn `A>=50` wahr ist bzw. zu `0`, wenn der Ausdruck falsch ist. `A` wird somit am `Then`-Pin angelegt, wenn der Ausdruck wahr ist, und `0` wird am `Else`-Pin angelegt, wenn der Ausdruck falsch ist. (Jeder beliebige Ausdruck, dessen Auswertung nicht Null ergibt, wird als wahr betrachtet, und der Wert wird an den `Then`-Pin weitergegeben.)

Testen von Zahlen, Schritt 3

Erstellen Sie eine Lösung mit nur drei Objekten.

Tipp: Verwenden Sie einen triadischen Ausdruck im `Formula`-Objekt. Das Format lautet: `(`*<Ausdruck>* `?` *<if TRUE,Ausgabewert>* `:` *<if FALSE, Ausgabewert>*`)`. Wenn beispielsweise `A < 10` in `TRUE` (wahr) ausgewertet wird, soll der Wert `A` am `Result`-Pin angelegt werden; andernfalls soll die Zeichenfolge “`FALSE`” am `Result`-Pin angelegt werden. Sie verwenden hierzu den folgenden triadischen Ausdruck: `(A<10 ? A : "FALSE")`.

Lösung—Testen von Zahlen, Schritt 3

Abbildung A-5 zeigt die Lösung zur Übung "Testen von Zahlen" mit nur drei Objekten.

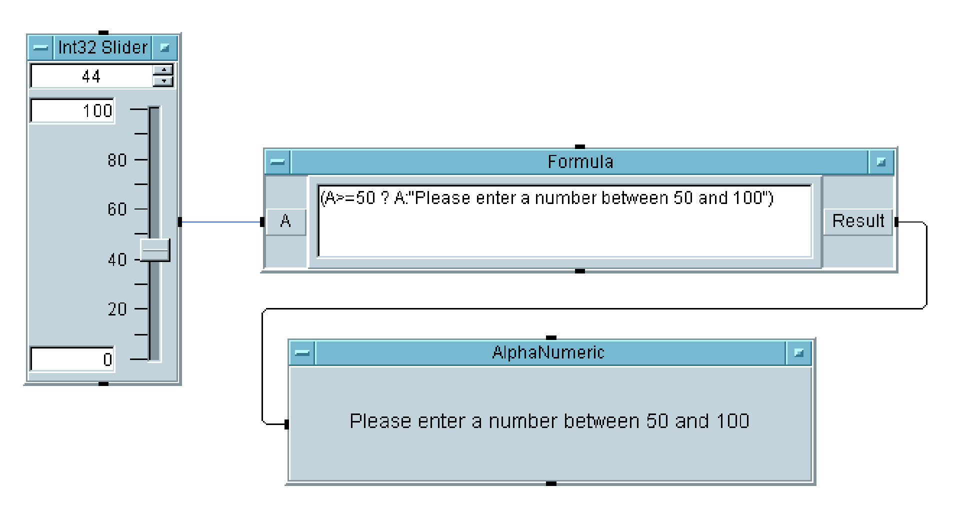


Abbildung A-5. Testen von Zahlen, Schritt 3

Hinweis

Dies kann über ein Dialogfenster `Real64 Input` mit der automatischen Fehlerprüfung implementiert werden. Der Bediener muss jedoch eine gültige Zahl eingeben, bevor das Programm abgeschlossen werden kann.

Erfassen von Zufallszahlen

Erstellen Sie ein Programm Create, das 100 Zufallszahlen generiert und anzeigt. Zeichnen Sie die erforderliche Gesamtzeit zum Generieren und Anzeigen der Werte an.

Vorschläge

Dieses Programm kann mit sechs oder weniger Objekten erstellt werden. Wählen Sie Objekte aus der folgenden Liste aus:

Start
For Range
Until Break
randomseed()-Funktion
random()-Funktion
Collector
Formula
Set Values
Alloc Int32
Logging AlphaNumeric
Strip Chart
Meter
Date/Time
Timer
Now()
Break
Do

Tipp

Senden Sie zum Verbessern der Leistung die Daten nur ein Mal an die Anzeige, indem Sie sie zunächst mit dem `Collector`-Objekt in einem Array erfassen. Beachten Sie die Unterschiede bei der Leistung.

Lösung—Erfassen von Zufallszahlen

Abbildung A-6 zeigt eine Lösung zur Übung "Erfassen von Zufallszahlen".

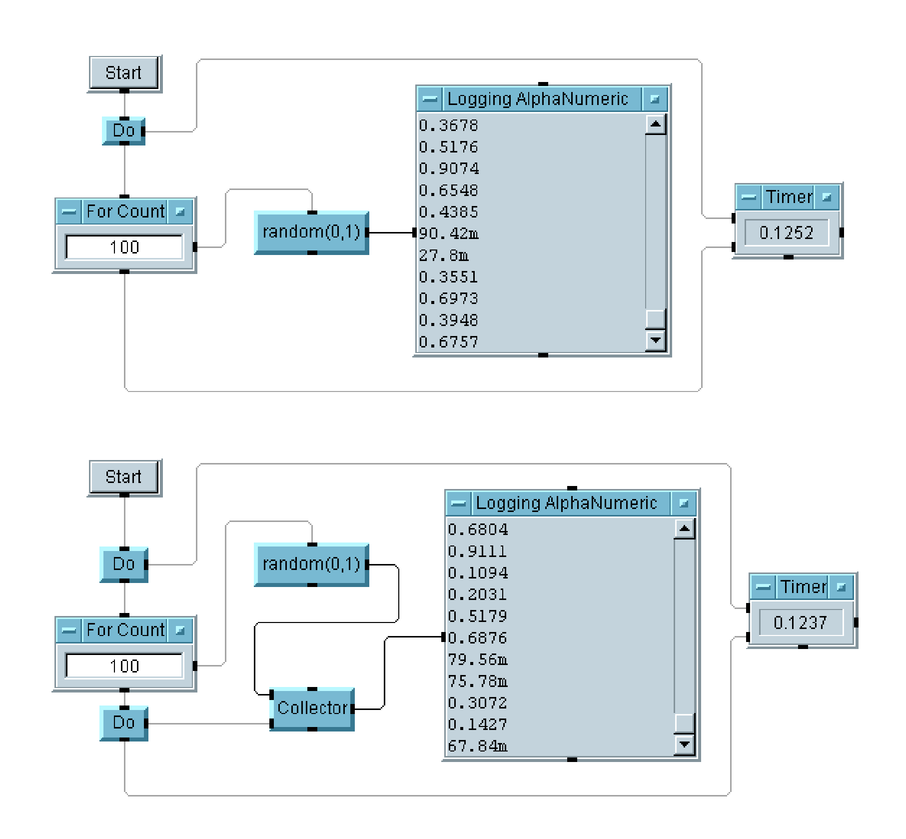


Abbildung A-6. Erfassen von Zufallszahlen

Wichtige Punkte

- **Logging AlphaNumeric im Vergleich zu AlphaNumeric:** Verwenden Sie `Logging AlphaNumeric` zum Anzeigen einer fortlaufenden Eingabe (entweder `Scalar` oder `Array 1D`) als eine Historie vorangegangener Werte. Verwenden Sie `AlphaNumeric` zum Anzeigen von Daten aus einer einzigen Ausführung (der letzten) als einzelnen Wert, als `Array 1D` oder als `Array 2D`. Die Anzeige `Logging` ist ein Array ohne Indexwerte; die Anzeige `AlphaNumeric` entspricht dem gleichen Array mit optionalen Indexnummern und Werten.

- **Timing-Pins:** Das `Do`-Objekt steuert, welches Objekt zuerst ausgeführt wird. Das Ende des Programms wird zeitlich gesteuert vom Sequenz-Ausgangs-Pin des Objekts `For Count`, da dieser Pin nicht ausgelöst wird, bis alle Objekte in der Schleife ausgeführt wurden.

Zufallszahlen-Generator

Zufallszahlen-Generator, Schritt 1

Erstellen Sie einen Zufallszahlen-Generator, der externe Eingänge erfordert. Zeigen Sie die Zahl in einem Streifendiagramm an. Eingänge sollen zulässig sein für:

Zufallszahl Maximum
Zufallszahl Minimum
Anzahl der generierten Zufallszahlen

Lösung—Zufallszahlen-Generator, Schritt 1

Abbildung A-7 zeigt eine Lösung für den ersten Schritt der Übung "Zufallszahlen-Generator".

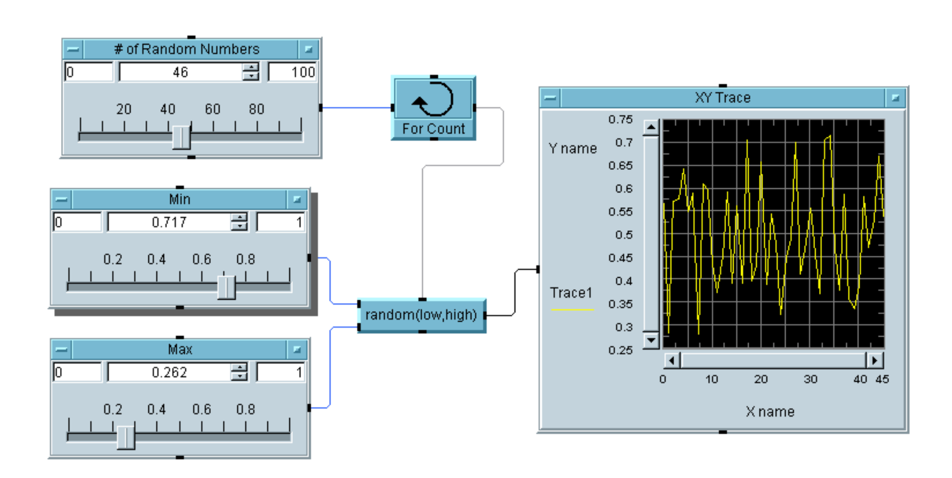


Abbildung A-7. Zufallszahlen-Generator, Schritt 1

Wichtige Punkte

- **Layout der Slider-Objekte:** Sie können für die Bildschirmanzeige der Slider-Objekte ein vertikales oder ein horizontales Format auswählen, indem Sie `Horizontal` unter `Layout` im Feld `Properties` anklicken.

- **XY Trace:** Verwenden Sie einen `XY Trace` zum Anzeigen der letzten Historie der laufend generierten Daten.

Zufallszahlen-Generator, Schritt 2

Erfassen Sie die Zufallszahlen in einem Array. Ermitteln Sie den laufend wechselnden Durchschnitt und zeigen Sie ihn mit den Zahlen an.

Lösung—Zufallszahlen-Generator, Schritt 2

Abbildung A-8 zeigt eine Lösung für den ersten Schritt der Übung "Zufallszahlen-Generator", Schritt 2.

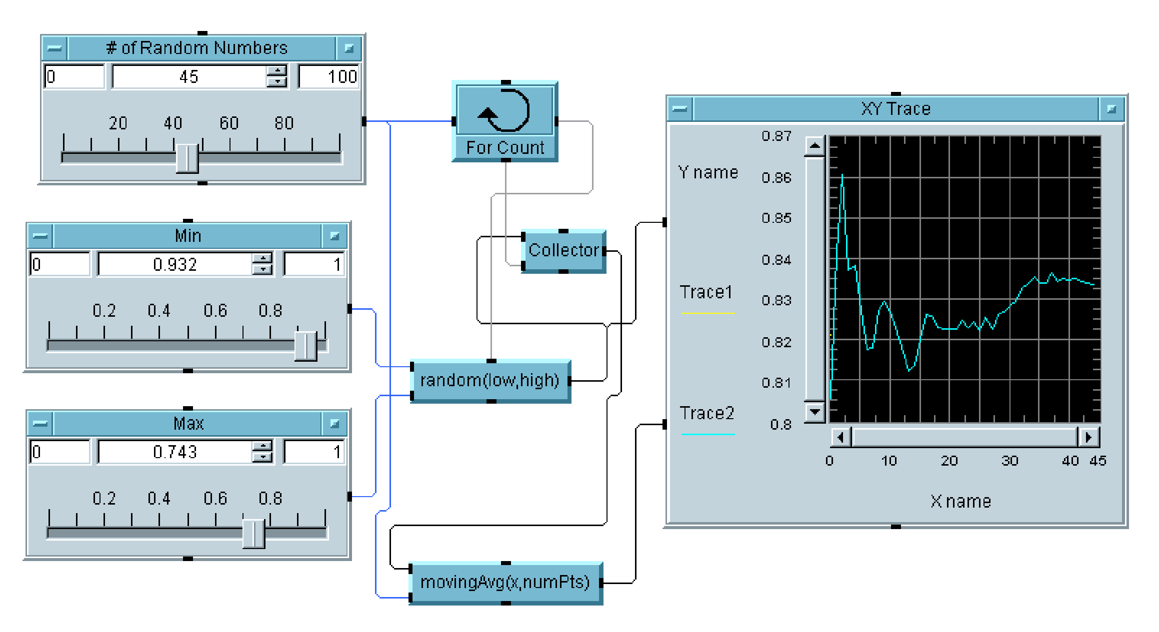


Abbildung A-8. Zufallszahlen-Generator, Schritt 2

- **MovingAvg(x, numPts):** Verwenden Sie dieses Objekt in der Kategorie `Function & Object Browser`, `Data Filtering` zum "Glätten" der Eingangsdaten mit dem Durchschnitt aus einer angegebenen Anzahl von Datenpunkten vor dem jeweiligen Punkt, um den "geglätteten" Datenpunkt zu berechnen.

Verwenden von Masken

Maskentest, Schritt 1

Erstellen Sie eine 50 Hz-Sinuskurve mit einem anpassbaren Störfaktor. Testen Sie die `gestörte Sinuskurve`, um sicherzustellen, dass die Kurve innerhalb der folgenden Grenzwerte verläuft:

(0,0.5)
(2.2m, 1.2)
(7.2m, 1.2)
(10.2m, 0.5)
(20m, 0.5)

Wenn die Sinuskurve die Grenzwerte überschreitet, markieren Sie die entsprechenden Punkte mit einem roten Karo.

Tipps

Sie können das Format der Anzeige von Linien über Punkte bis zu Karos ändern. (Wählen Sie in `Properties` das Register `Traces` für jeden Trace-Eingang; der Linientyp kann durchgezogen, gestrichelt, gepunktet etc. sein. Als Punkttyp können Punkt, Karo, Kästchen oder andere Formen verwendet werden.) Das Objekt `Comparator` kann hierbei sehr nützlich sein.

Lösung—Verwenden von Masken, Schritt 1

Abbildung A-9 zeigt eine Lösung für Schritt 1.

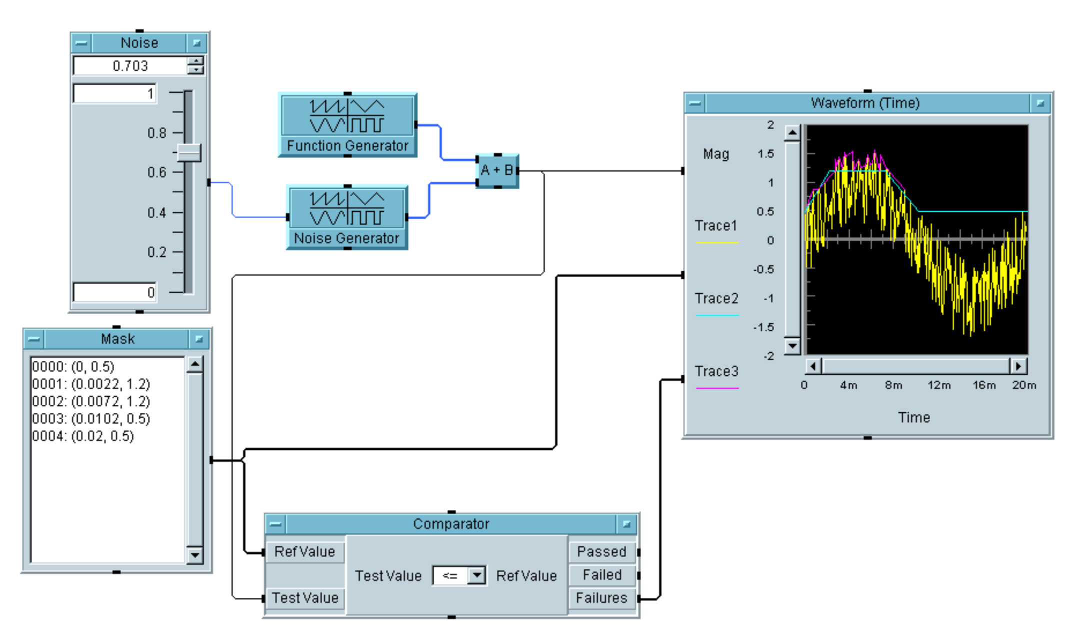


Abbildung A-9. Der Maskentest, Schritt 1

Verwenden von Masken, Schritt 2

Erweitern Sie das Programm, sodass es den Prozentanteil der Fehler berechnet und anzeigt.

Lösung—Verwenden von Masken, Schritt 2

Abbildung A-10 zeigt eine Lösung für Schritt 2.

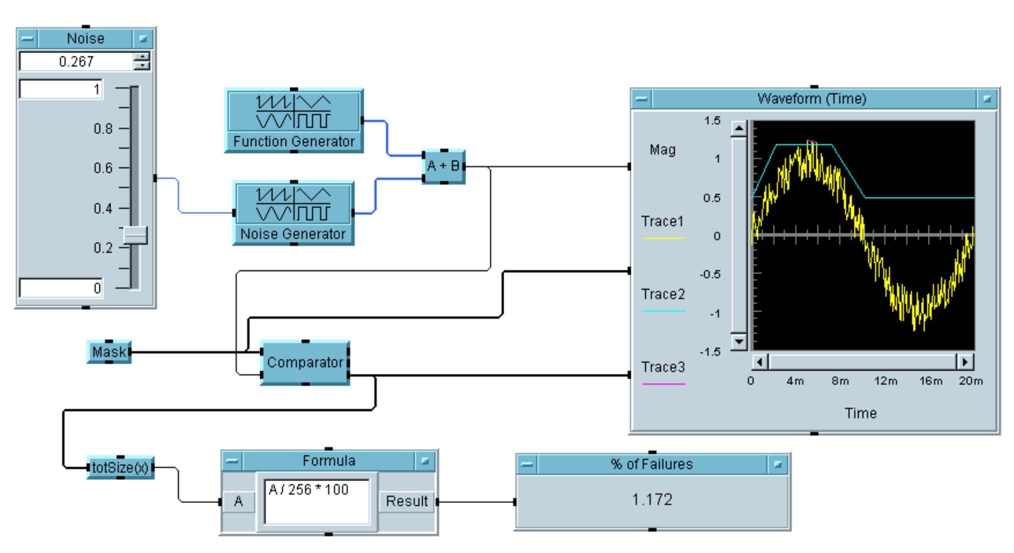


Abbildung A-10. Der Maskentest, Schritt 2

Wichtige Punkte

- **Maske:** Die Maske wird mit dem Objekt `Data` ⇒ `Constant` ⇒ `Coord` erstellt und anschließend für fünf Array-Elemente konfiguriert. Sie geben die Koordinatenpaare durch Kommas getrennt ein; VEE fügt die Klammern hinzu. Bei der Auswahl der `x`-Werte war bekannt, dass die Zeitdauer der Wellenform 20 Millisekunden betrug. Beachten Sie auch, dass die Anzeige `Waveform (Time)` den Datentyp `Coord` als Eingang akzeptiert. Sie könnten auch ein Objekt `Data` ⇒ `Build Data` ⇒ `Arb Waveform` verwenden; dieses Objekt wandelt einen Datentyp `Coord` in einen Datentyp `Waveform` um durch Angabe der Anzahl von Punkten in der `Waveform`.

- **Comparator:** Dieses Objekt vergleicht einen Testwert mit einem Referenzwert. Auch hierbei können Sie eine Wellenform mit einem Array aus Koordinatenpaaren vergleichen. Der `Failures`-Pin gibt einen Array der fehlgeschlagenen Datenpunkte aus; Sie können diesen Array an die Anzeige senden und mit einer anderen Farbe oder einem anderen Linientyp hervorheben.

- **TotSize:** Dieses Objekt gibt einfach die Anzahl der Elemente in einem Array aus. Da dieser Array die Anzahl der Fehler enthält, können Sie den Prozentanteil der Fehler ermitteln, indem Sie diesen Wert durch die Anzahl der Elemente in der ursprünglichen Wellenform (256) dividieren und das Ergebnis mit 100 multiplizieren.

- **Formula:** `A/256*100` ist die zur Berechnung des Fehler-Prozentanteils verwendete Formel, da der `Function Generator` und der `Noise Generator` auf 256 Punkte eingestellt sind.

Verwenden von Zeichenfolgen und globalen Variablen

Manipulieren von Zeichenfolgen und globalen Variablen

Erstellen Sie mit Objekten oder Funktionen ein Programm, das einen Benutzernamen im folgenden Format akzeptiert: *<Leerzeichen>* *<Vorname>* *<Leerzeichen>* *<Nachname>*. Nachdem der Benutzer einen Namen eingegeben hat, soll das Programm den Vornamen weglassen und nur den Nachnamen ausgeben. Speichern Sie die Zeichenfolge in einer globalen Variablen. Rufen Sie die Zeichenfolge mit dem `Formula`-Objekt ab.

Lösung—Manipulieren von Zeichenfolgen und globalen Variablen

Abbildung A-11 zeigt eine Lösung zur Übung "Manipulieren von Zeichenfolgen und globalen Variablen".

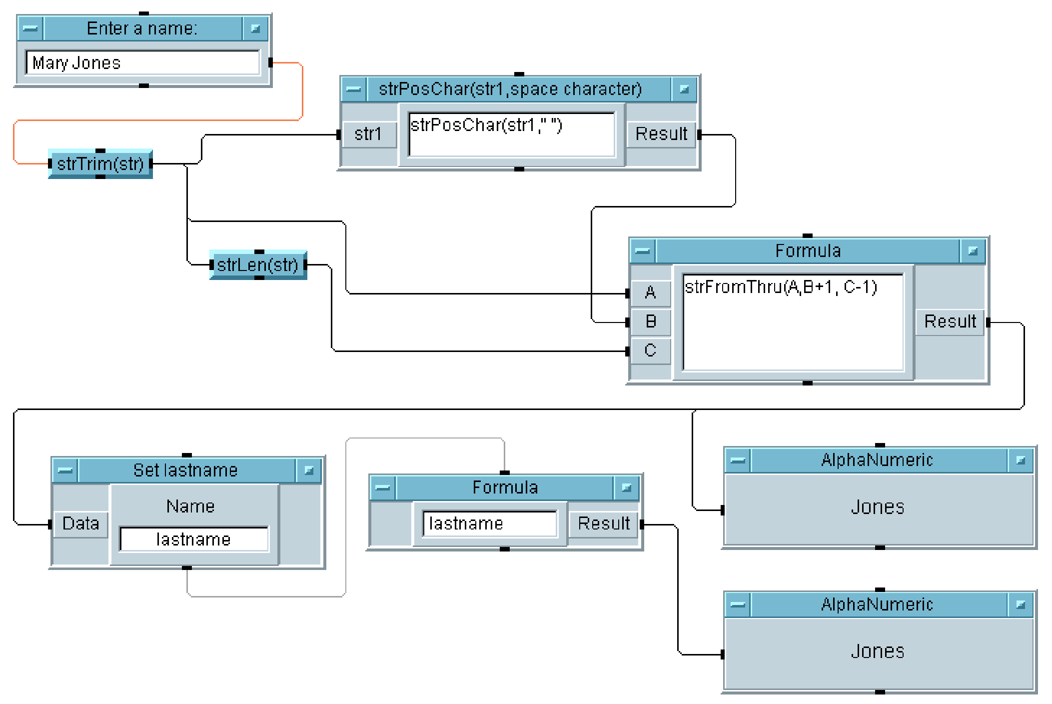


Abbildung A-11. Manipulieren von Zeichenfolgen und globalen Variablen

Wichtige Punkte

- **Zeichenfolgeobjekte und Funktionen:** `StrTrim(str)` entfernt zunächst alle Leerzeichen und Tabulatorzeichen vor oder hinter dem Namen. `StrPosChar(str1, " ")` liefert den Index des Leerzeichens zwischen dem `firstname` und dem `lastname`. `StrLen(str)` liefert die Länge der Zeichenfolge. Alle diese Funktionen wurden mit den Zeichenfolgeobjekten ausgeführt; Sie könnten jedoch auch mit Zeichenfolgefunktionen innerhalb eines `Formula` -Objekts ausgeführt werden.

- **Formelobjekt:** `StrFromThru(A,B+1,C-1)` verwendet den Zeichenfolgeeingang `A`, addiert `1` zu dem Index des Leerzeichens von Eingang `B` und subtrahiert `1` von der Länge der Zeichenfolge am Eingang `C`. (Die gesamte Indexierung erfolgt zur Basis Null.)

- **Set Variable:** Die globale Variable `lastname` kann sehr einfach gesetzt und anschließend in einem beliebigen Ausdrucksfeld referenziert werden; in diesem Beispiel etwa im `Formula`-Objekt.

- **Optimieren:** Die drei Formeln können zu einer einzigen Formel kombiniert werden. Es empfiehlt sich, `strTrim()` als separates Element beizubehalten, da sein Ausgang mehrmals verwendet wird; die anderen Elemente können jedoch kombiniert werden, um die Geschwindigkeit zu optimieren. (Das Programm ist dadurch allerdings etwas weniger leicht lesbar.)

Optimierungstechniken

Für diese Übung erstellen Sie ein VEE-Programm auf zwei verschiedene Arten. Beachten Sie den Unterschied bei der Ausführungsgeschwindigkeit.

Optimierungstechniken, Schritt 1

Erstellen Sie ein Programm, das den Bereich `0 to 710 step 10` über eine Sinus- und eine Kosinusfunktion sendet. Leiten Sie das Ergebnis dieser Funktionen an eine Anzeige `X vs.Y`. Verwenden Sie das Objekt `Timer`, um zu messen, wie lang die Ausführung des Programms dauert. (Legen Sie die Standardeinstellungen für `Trig Mode` auf `Radians` fest.)

Lösung—Optimierungstechniken, Schritt 1

Abbildung A-12 zeigt eine Lösung für Schritt 1.

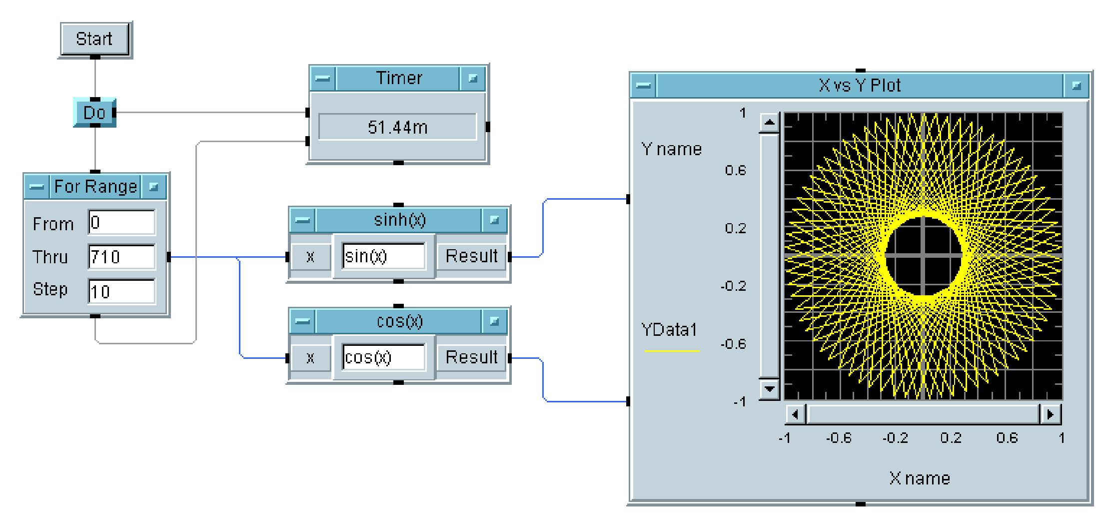


Abbildung A-12. Optimieren von VEE-Programmen, Schritt 1

Optimierungstechniken, Schritt 2

Klonen Sie alle Objekte aus dem ersten Programm. Ändern Sie die neue Gruppe so, dass sie den Bereich in einem Array erfasst. Die Funktionen `sine` und `cosine` werden jetzt mit einem Array aus Punkten ausgeführt und nur ein einziges Mal geplottet. Beachten Sie die Zeitersparnis.

Lösung—Optimierungstechniken, Schritt 2

Abbildung A-13 zeigt eine Lösung für Schritt 2.

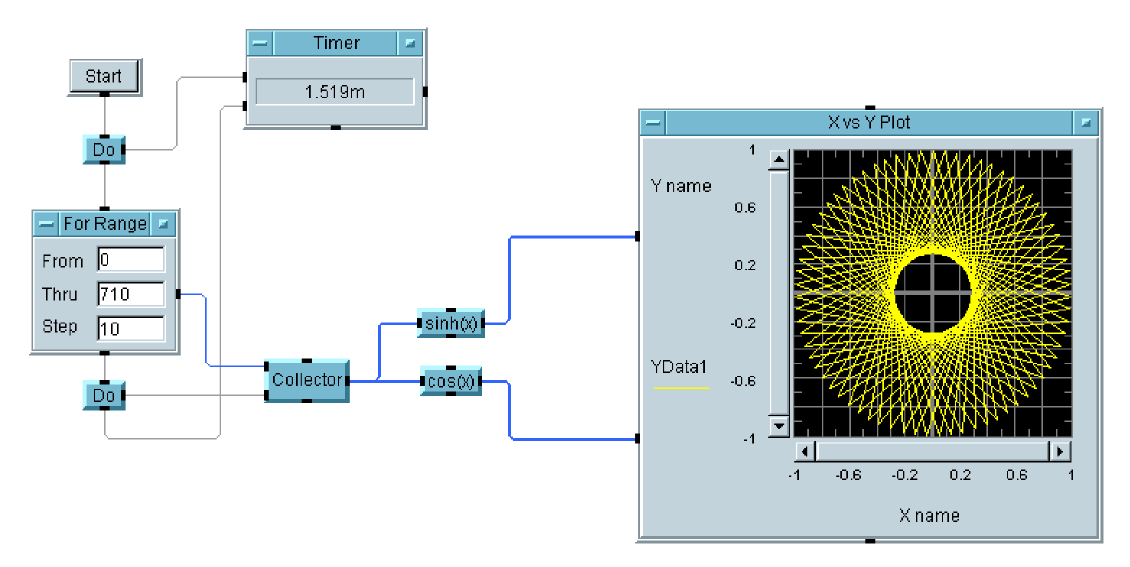


Abbildung A-13. Optimieren von VEE-Programmen, Schritt 2

Wichtige Punkte

- **Optimieren mit Arrays:** Beachten Sie die Verbesserung der Leistung von Schritt 1 auf Schritt 2 durch die Verwendung von Arrays. Führen Sie die Analyse oder Anzeige von Ergebnissen nach Möglichkeit immer mit `Arrays` statt mit `Skalarwerten` durch.

- **X gg. Y-Anzeige:** Dieses Beispiel verwendet diese Anzeige statt der Anzeigen `Waveform` oder `XY`, da separate Daten für `X` und `Y` vorliegen.

UserObjects

UserObject "Random Noise"

UserObject "Random Noise", Schritt 1

Erstellen Sie ein `UserObject`, das eine "Random Noise"-Waveform (Wellenform mit Zufallsstörung) generiert. Zeigen Sie die gestörte Wellenform und das Störungsspektrum außerhalb des `UserObject` an. Ermöglichen Sie die Steuerung folgender Komponenten außerhalb des `UserObject`: `amplitude`, `number of points` (Anzahl der Punkte), `interval (time span)` (Intervall), `DC offset` (Gleichstromversatz).

Hinweis

Verwenden Sie keine virtuelle Quelle innerhalb des `UserObject`. Verwenden Sie Objekte wie beispielsweise `Build Waveform` und `Random` zum Erstellen des `UserObject`.

Lösung—UserObject "Random Noise"

Abbildung A-14 zeigt eine Lösung für das `UserObject "Random Noise"`.

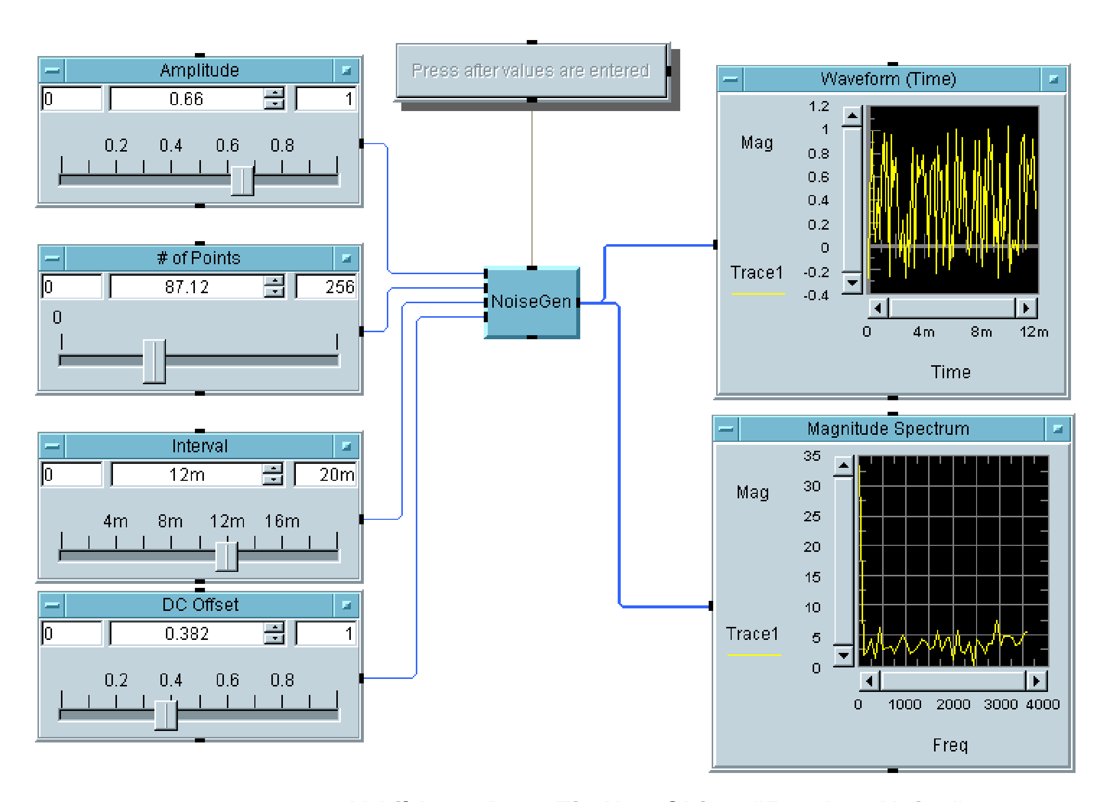


Abbildung A-14. Ein UserObject "Random Noise"

Lösung—NoiseGen-Objekt in Random Noise

Abbildung A-15 zeigt eine Lösung für das `UserObject NoiseGen`.

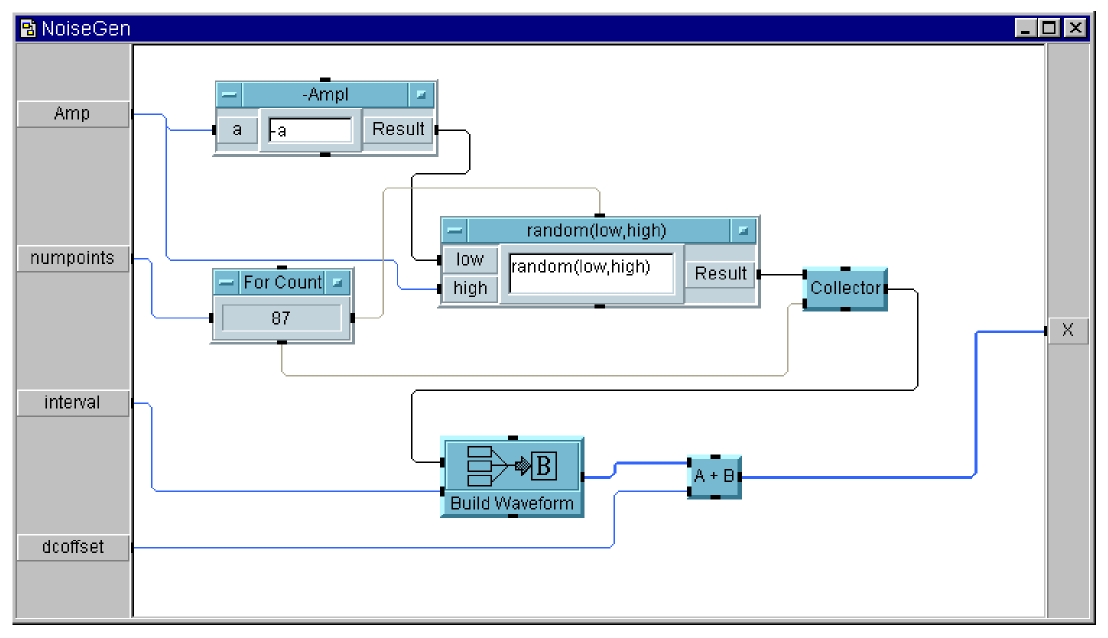


Abbildung A-15. Das UserObject "NoiseGen"

Wichtige Punkte

- **UserObject:** Beachten Sie, dass die von Ihnen erstellten `UserObjects` im Grund angepasste Objekte sind, die Sie VEE hinzufügen.
- **Build Waveform:** Dieses Objekt erstellt einen Datentyp `Waveform` aus einem `Real`-Array von Amplitudenwerten und einer `time span` (der Zeitdauer in Sekunden, über die die `y`-Daten gesampelt wurden).

Agilent VEE UserFunctions

Verwenden von UserFunctions

UserFunctions, Schritt 1

Erstellen Sie eine Funktion `NoiseGen`, die einen Amplitudenwert (0-1) aus einem Slider-Objekt akzeptiert und eine gestörte Wellenform zurückgibt.

Verwenden Sie folgende Komponenten NICHT:
Virtual Source
For Count
For Range

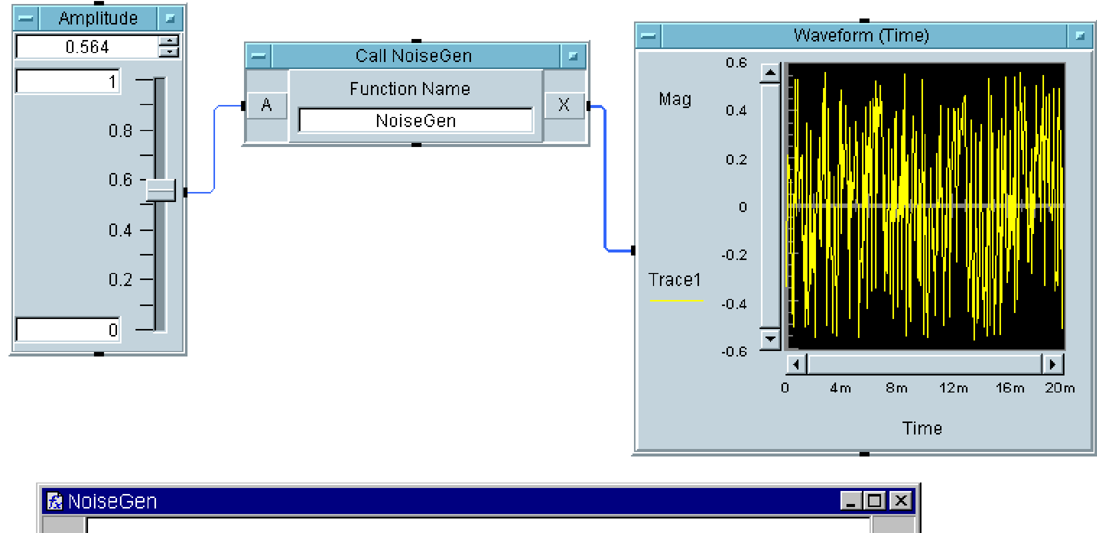
Verwenden Sie:
Formula
Ramp
Build Waveform

Tipp

Verwenden Sie `randomize(array, -a,a)`, wobei der Array 256 Punkte haben muss und `a` die Amplitude ist. Erstellen Sie ein einfaches Hauptprogramm, das diese Funktion aufruft; auf diese Weise können Sie sicherstellen, dass die Funktion richtig funktioniert.

Lösung—UserFunctions, Schritt 1

Abbildung A-16 zeigt eine Lösung für Schritt 1.



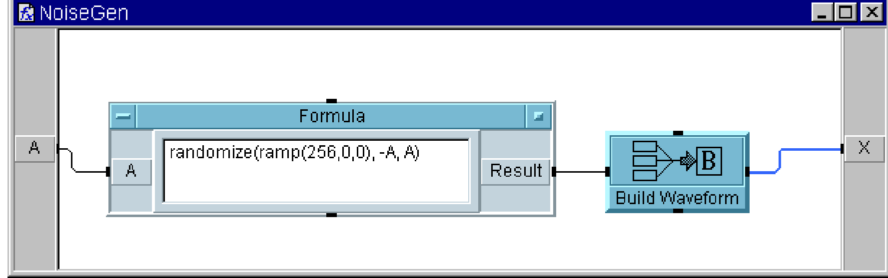


Abbildung A-16. UserFunctions, Schritt 1

Wichtige Punkte

- **Ramp():** Beachten Sie, dass mit der Funktion `ramp()` innerhalb der Parameterliste für `randomize()` ein Array aus 256 Punkten generiert wird.

- **Build Waveform:** Beachten Sie, dass die Standard-Zeitdauer ("default time span") 20 Millisekunden beträgt; Sie brauchen daher nur einen Array an dieses Objekt zu senden, um eine Wellenform zu generieren.

UserFunctions, Schritt 2

Erstellen Sie in dem gleichen Programm eine weitere Funktion `AddNoise`, die die erste Funktion `NoiseGen` aufruft. `AddNoise` sollte die gestörte Wellenform von der Funktion `NoiseGen` zu einer Sinuskurve addieren. `AddNoise` sollte zwei Eingänge haben, einen für die NoiseGen-Amplitude und einen für die Sinuskurve. Sie sollte einen Ausgang für das Ergebnis haben.

Erstellen Sie ein einfaches Hauptprogramm mit einem Slider für die Noise-Amplitude und die `Virtual Source⇒ Function Generator` (`sine wave`, `Freq = 100 Hz`) für die gute Wellenform, zu der die Störung addiert werden soll. Zeigen Sie die resultierende Wellenform an.

Lösung—UserFunctions, Schritt 2

Abbildung A-17 zeigt eine Lösung für Schritt 2.

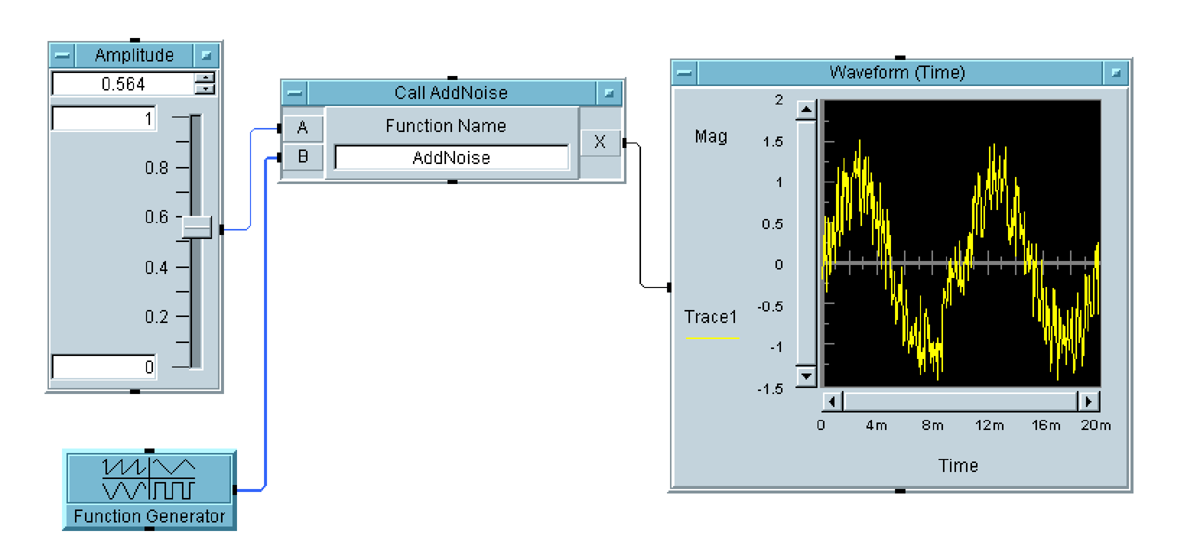


Abbildung A-17. UserFunctions, Schritt 2

UserFunctions, Schritt 3

Rufen Sie im gleichen Programm die Funktion `AddNoise` erneut auf; dieses Mal von einem `Formula`-Objekt aus, und verwenden Sie den absoluten Wert des Ergebnisses. Zeigen Sie die Wellenform für den absoluten Wert in der gleichen Anzeige an. Bereiten Sie als nächstes die Bearbeitung der Funktion `AddNoise` vor. Schalten Sie `Debug` ⇒ `Show Data Flow` ein. Lassen Sie das Fenster `AddNoise` geöffnet und führen Sie das Programm aus. Beachten Sie, wie nützlich diese Funktion für die Fehlerbehebung ist.

Lösung—UserFunctions, Schritt 3

Abbildung A-18 zeigt eine Lösung für Schritt 3.

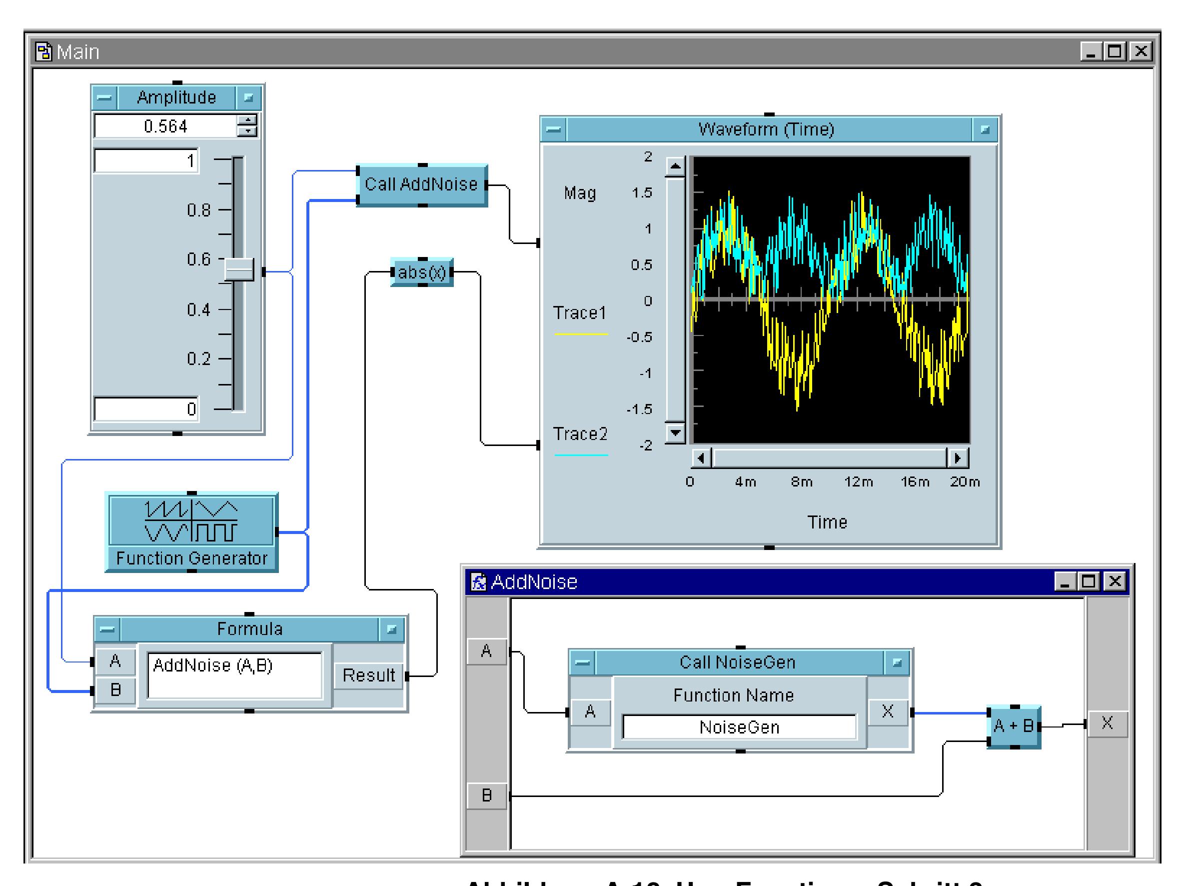


Abbildung A-18. UserFunctions, Schritt 3

UserFunctions, Schritt 4

Ändern Sie jetzt das Programm so, dass der Slider eine globale Variable mit dem Namen `Amplitude` festlegt. Sorgen Sie dafür, dass die Funktion `NoiseGen` diese globale Variable verwendet (damit NoiseGen keinen Eingangs-Pin mehr erfordert). Stellen Sie sicher, dass das Programm richtig ausgeführt wird. Speichern Sie diese Datei als `uflab.vee`.

Lösung—Verwenden von UserFunctions, Schritt 4

Abbildung A-19 zeigt eine Lösung für Schritt 4.

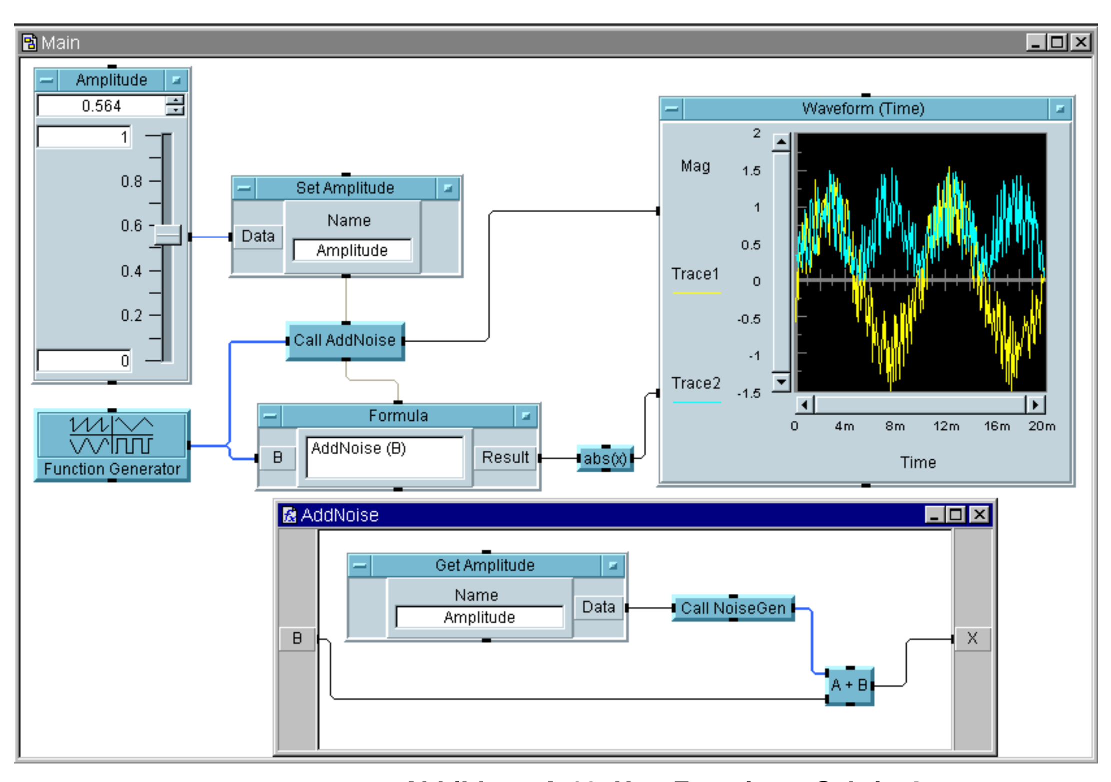


Abbildung A-19. UserFunctions, Schritt 4

Tipp: Beachten Sie, dass die Objekte `Call AddNoise` und `Formula` die globale Variable `Amplitude` verwenden; daher müssen beide Objekte nach Ausführung des Objekts `Set Amplitude` ausgeführt werden. Durch das Verbinden der Sequenz-Pins von `Set Amplitude` mit `Call AddNoise` und von `Call AddNoise` mit `Formula` wird sichergestellt, dass die Objekte in der richtigen Reihenfolge ausgeführt werden.

Importieren und Löschen von Bibliotheken von UserFunctions

Erstellen Sie ein einfaches Programm zum Importieren der Funktionen `uflab.vee` aus der vorigen Übung. Rufen Sie die Funktion auf, die die Störung hinzufügt, und löschen Sie anschließend die Bibliothek über das Programm. Verwenden Sie die Auswahl `Select Function` im Objektmenü des Objekts `Call`.

Tipp: Klicken Sie `Load Lib` im `Import Library`-Objektmenü an, um die angegebene Bibliothek manuell zu laden, damit Sie die Funktion `Select Function` in `Call` verwenden können.

Lösung—Importieren und Löschen von Bibliotheken über ein Programm

Abbildung A-20 zeigt eine Lösung zum programmgesteuerten Löschen der Bibliothek.

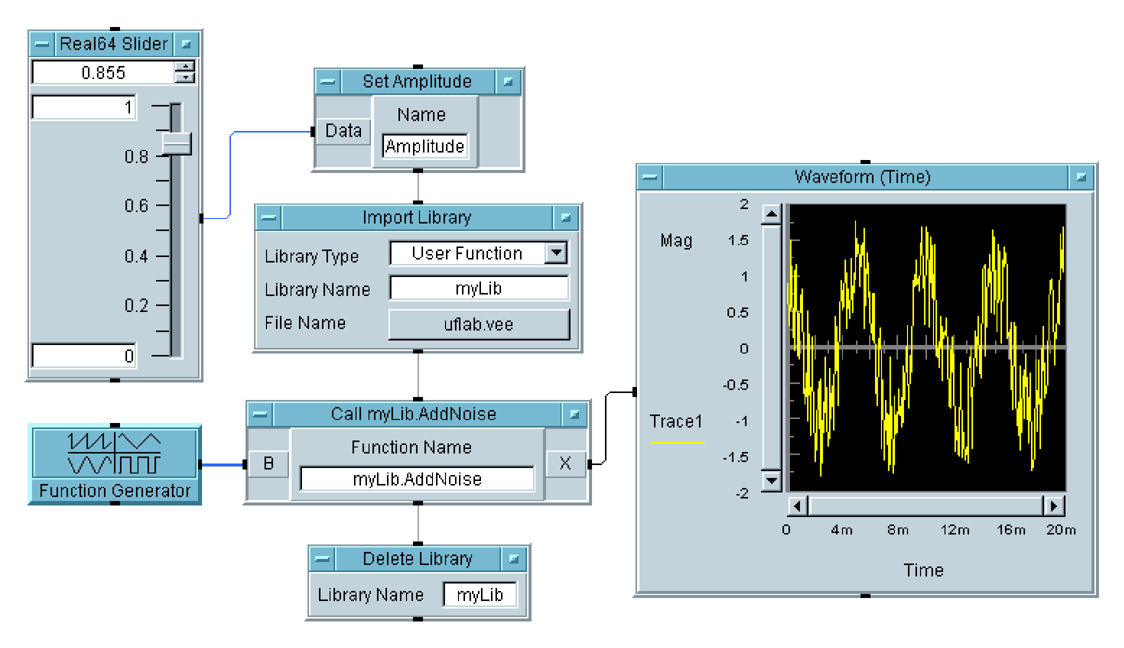


Abbildung A-20. Importieren und Löschen von Bibliotheken

Wichtige Punkte

- **Select Function:** Beachten Sie, dass diese Auswahl die richtigen Ein- und Ausgangs-Pins für die ausgewählte Funktion konfiguriert.
- **UserFunctions bearbeiten:** Wenn Sie eine Bibliothek von `UserFunctions` über ein Programm importieren, können Sie die Funktionen nicht bearbeiten. Sie können die Funktionen anzeigen und Unterbrechungspunkte für die Fehlerbehebung setzen. Wenn Sie die importierten `UserFunctions` bearbeiten wollen, verwenden Sie den Befehl `Merge Library`.
- **Variable "Caution" setzen:** Bei Verwendung einer globalen Variablen in einer Funktion müssen Sie daran denken, diese globale Variable zu erstellen, wenn Sie die Funktion in anderen Programmen verwenden wollen. Einer der Vorteile beim expliziten Erstellen von Eingängen und Ausgängen ist, dass diese einfacher nachzuvollziehen sind.

Erstellen von Bediener- und Einblend-fenstern

Erstellen von Bediener- und Einblendfenstern, Schritt 1

Erstellen Sie ein Fenster, über das der Bediener aufgefordert wird, Zahlen einzugeben. Erstellen Sie ein `UserObject` für die Interaktion mit einem Bediener. Fordern Sie den Bediener zu 2 Eingaben auf, `A` und `B`. Senden Sie beide Eingaben an eine Anzeige. Verwenden Sie ein `UserObject` mit ausgewählter Option `Show On Execute` zum Anzeigen des Fensters.

Lösung—Erstellen der Bediener- und Einblendfenster, Schritt 1

Abbildung A-21 zeigt eine Lösung in der Detailansicht. Abbildung A-22 zeigt das Fenster, das bei der Ausführung des Programms erscheint.

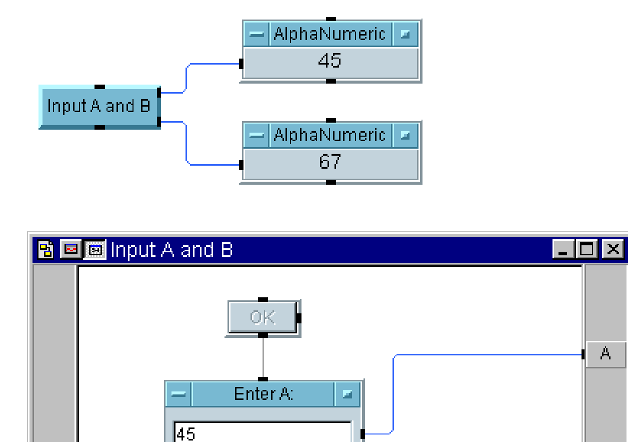


Abbildung A-21. UserObject, das den Bediener zur Eingabe von A und B auffordert

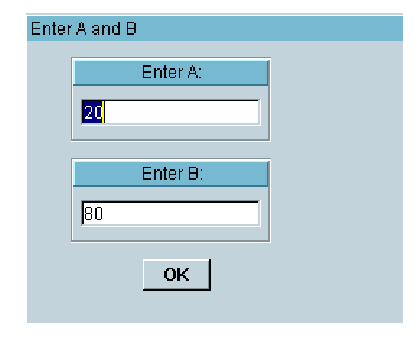


Abbildung A-22. Fenster zur Eingabe von A und B durch den Bediener

Wichtige Punkte

- **UserObject Properties:** Wählen Sie im Dialogfenster `UserObject Properties` die Option `Pop-Up Panel` aus und klicken Sie, um `Show Panel On Execute` einzuschalten. Ändern Sie den Namen `Pop-Up Panel` ⇒ `Panel Title` in “`Enter A or B`.”

Erstellen von Bediener- und Einblendfenstern, Schritt 2

Statt sowohl `A` als auch `B` anzuzeigen, fragen Sie den Bediener, ob `A` oder `B` angezeigt werden soll, wenn die beiden Zahlen unterschiedlich sind. Wenn nach der Abfrage die beiden Werte `A` und `B` gleich sind, zeigen Sie den Wert an. Sind die beiden Werte `A` und `B` unterschiedlich, fordern Sie den Bediener auf, einen Wert für die Anzeige auszuwählen. Zeigen Sie je nach der Auswahl des Bedieners `A` oder `B` an.

TIPP: Fügen Sie ein weiteres `UserObject` mit einem Einblendfenster mit der Einstellung `Show Panel on Execute` hinzu, und fordern Sie den Bediener auf, den Wert hier einzugeben.

Lösung—Erstellen der Bediener- und Einblendfenster, Schritt 2

Abbildung A-23 zeigt das `UserObject`, das den Bediener auffordert, eine Auswahl zu treffen, wenn `A` und `B` unterschiedliche Zahlen sind. Abbildung A-24 zeigt das zweite Einblendfenster, über das der Bediener gefragt wird, ob `A` oder `B` angezeigt werden soll.

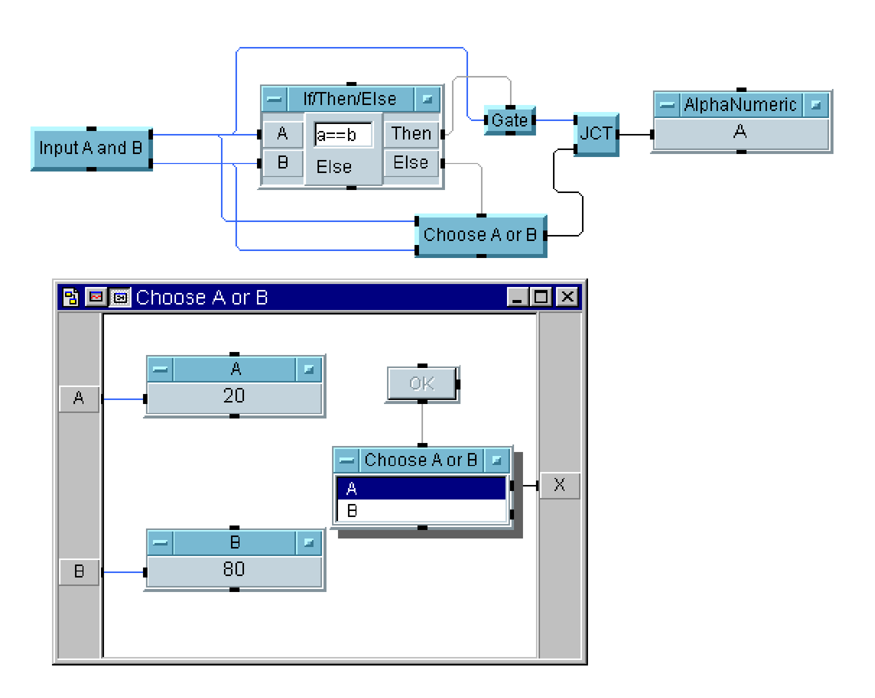


Abbildung A-23. UserObject, das den Bediener fragt, ob A oder B angezeigt werden soll

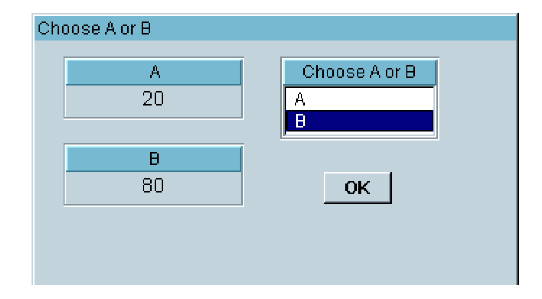


Abbildung A-24. Fenster für Bediener zur Auswahl, ob A oder B angezeigt werden soll

Wichtige Punkte

- **Gate:** Das Objekt `Gate` sendet nur einen Wert, wenn die beiden Zahlen gleich sind.
- **Junction:** Das Objekt `JCT` ermöglicht mehrfache Eingänge zum Objekt `Alphanumeric`. Das Objekt `JCT` ist ein "Verdrahtetes ODER"-Objekt.
- **Objekt als Menü auflisten:** Beachten Sie die Verwendung des Objekts `Data ⇒ Selection Controls ⇒ List` für zwei Auswahloptionen und die Formatierung für eine Liste. Diese Konfiguration gibt einen Text `A` oder `B` aus. Wenn Sie den Ordnungswert (`0` oder `1`) benötigen, verwenden Sie stattdessen den Ordnungs-Datenausgang des Objekts `List`.

Erstellen von Bediener- und Einblendfenstern, Schritt 3

Wenn der Bediener keine Zahlen eingibt, soll eine Fehlermeldung generiert werden. Fügen Sie beim zweiten `UserObject`, das den Bediener bei zwei unterschiedlichen Zahlen zur Auswahl zwischen `A` und `B` für die die Anzeige auffordert, einen Fehler hinzu. Wenn der Bediener innerhalb von 10 Sekunden weder `A` noch `B` auswählt, soll der Fehler generiert werden.

Lösung—Erstellen der Bediener- und Einblendfenster, Schritt 3

Abbildung A-25 zeigt das `UserObject` mit der Änderung, sodass es einen Fehler generiert, wenn der Bediener nicht innerhalb von 10 Sekunden `A` oder `B` auswählt.

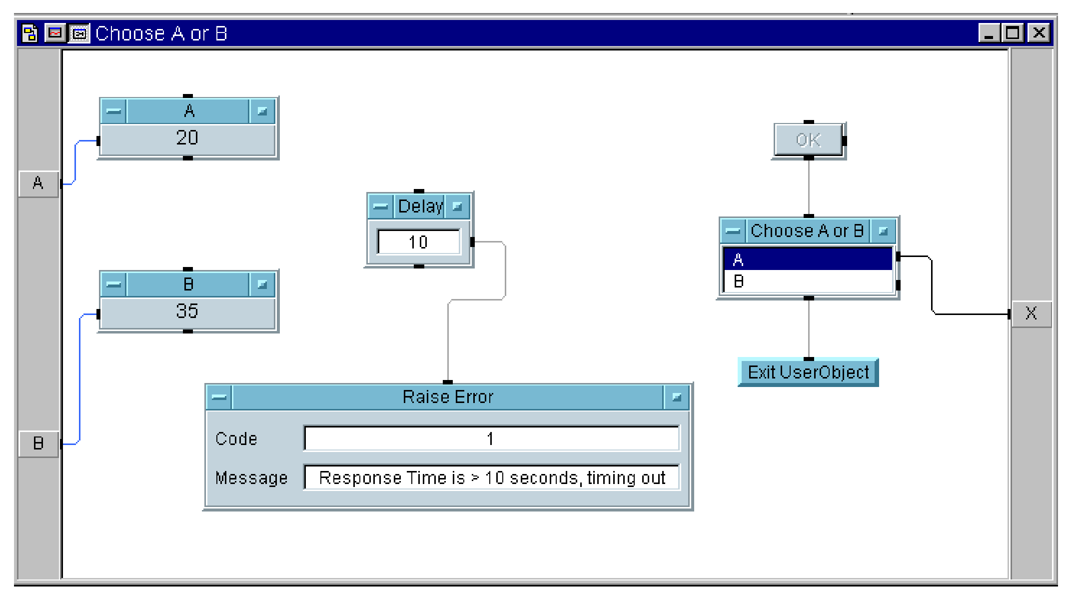


Abbildung A-25. Fehler generieren, wenn der Bediener keine Auswahl eingibt

Wichtige Punkte

- **UserObject beenden:** Wenn der Bediener vor Ablauf von 10 Sekunden reagiert, beendet dieses Objekt das `UserObject`, auch wenn die Ausführung des Objekts `Delay` noch nicht abgeschlossen ist.

- **Verzögerung und Fehler auslösen:** Nach 10 Sekunden führt das Objekt `Delay` einen "Ping" auf das Objekt `Raise Error` aus; dadurch wird die Ausführung des Programms angehalten und die eingegebene `Fehlermeldung` angezeigt. Außerdem wird ein roter Umriss um das Objekt angezeigt, das den Fehler verursacht hat; diese rote Umrisslinie verschwindet, wenn eine der Schaltflächen **Stopp** oder **Ausführen** in der Hauptmenüleiste angeklickt wird.

- **OK und Verzögerung:** Beachten Sie, dass die beiden Threads in `AnotB` getrennt sind, sodass `OK` und `Delay` gleichzeitig ablaufen.

Arbeiten mit Dateien

Verschieben von Daten in und aus Dateien

Erstellen Sie ein VEE-Programm, das die Uhrzeit in eine Datei schreibt. Generieren Sie 100 Zufallspunkte und schreiben Sie sie in die Datei. Berechnen Sie den Mittelwert und die Standardabweichung der Zahlen und hängen Sie sie im folgenden Format an die Datei an:

Mittelwert: xxxxxx
Std.Abweichung: yyyyyy

Lesen Sie als nächstes nur den Mittelwert und die Standardabweichung aus der Datei aus. Abbildung A-26 zeigt das Verschieben von Daten in und aus Dateien.

Lösung—Verschieben von Daten in und aus Dateien

Abbildung A-26 zeigt eine Lösung zum Verschieben von Daten in und aus Dateien.

.

Abbildung A-26. Verschieben von Daten in und aus Dateien

Wichtige Punkte

- **Generieren eines Array:** Verwenden Sie `randomize(ramp(100,0,1), 0, 1)` im `Formula`-Objekt zum Erstellen eines Array aus 100 Zufallszahlen. Die Funktion `ramp()` generiert einen geordneten Array und liefert ihn an die Funktion `randomize()`, die dann Zufallswerte zwischen `0` und `1` generiert.

- **Zeitstempel:** Die Funktion `now()` wird im Ausdrucksfeld des Dialogfensters `I/O Transaction` für die Transaktion Eins des Objekts `To File` verwendet. Wenn Sie das Format in `TIME STAMP FORMAT` ändern, werden zusätzlichen Schaltflächen in dem Dialogfenster angezeigt, um anzugeben, wie die Uhrzeit gespeichert werden soll.

- **Speichern von zwei Werten in einer Zeile:** In der dritten und vierten Transaktion des Objekts `To File` wird eine konstante `Text`zeichenfolge gespeichert, gefolgt von einem `Real`-Wert. In der dritten Transaktion geben Sie beispielsweise `"Mean: " B` im Ausdrucksfeld des Fensters `I/O Transaction` ein (sofern der Mittelwert am Eingangs-Pin `B` anliegt).

- **Extrahieren eines Werts aus einer Datei:** Zum Ermitteln des Mittelwerts und der Standardabweichung senden Sie zunächst einen Befehl `EXECUTE REWIND`, um den Lesezeiger an den Anfang zu positionieren. Verwenden Sie anschließend `NULL` mit dem richtigen Format, um eine `READ`-Transaktion über den Zeitstempel und den Real-Array hinaus auszuführen. (Dadurch werden die eingelesenen Werte gelöscht, statt sie in einem Ausgangsanschluss abzulegen.) Schließlich lesen Sie die beiden letzten Zeilen in der Datei als Zeichenfolgen ein.

- **Junction:** Verwenden Sie das Objekt `Flow ⇒ Junction`, um mehrere Ausgänge zu einem einzigen Eingang zu verbinden, beispielsweise zum Verbinden der Ausgänge `mean` und `sdev` zur Anzeige `Logging AlphaNumeric`.

Datensätze

Manipulieren von Datensätzen

Manipulieren von Datensätzen, Schritt 1

Erstellen Sie einen Datensatz mit drei Feldern, die eine Ganzzahl enthalten, die aktuelle Uhrzeit als Zeichenfolge und einen Array aus vier Real-Elementen. Die Felder sollen die Namen `int`, `daytime` und `rarry` haben. Fügen Sie diesen Datensatz mit einem anderen, der eine Zufallszahl zwischen 0 und 1 enthält, und einer Wellenform zusammen. Benennen Sie diese Felder `rand` und `wave`.

Lösung—Manipulieren von Datensätzen, Schritt 1

Der resultierende Datensatz sollte fünf Felder enthalten, wie in Abbildung A-27 gezeigt.

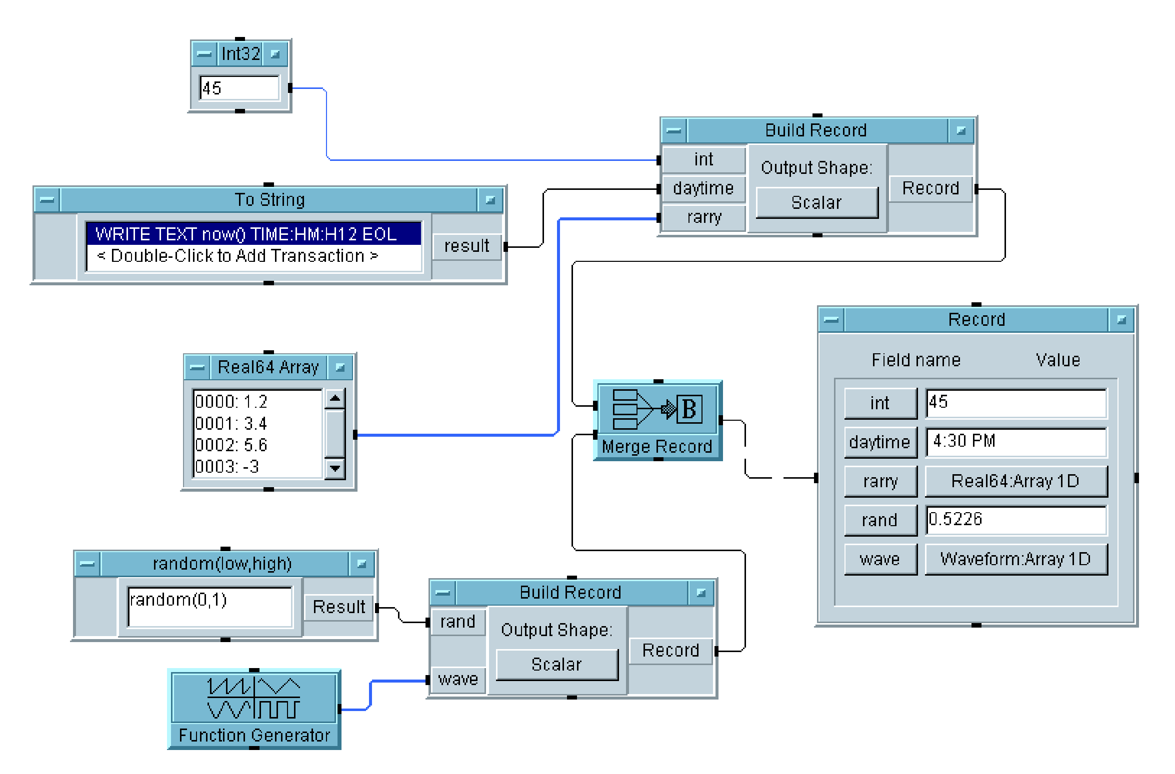


Abbildung A-27. Manipulieren von Datensätzen, Schritt 1

Wichtige Punkte

- **Zeitstempel:** Verwenden Sie die Funktion `now()` im Objekt `To String`, um Ihren Zeitstempel für dieses Programm zu erstellen. Anschließend können Sie das Format angeben.

- **Konfigurieren einer Datenkonstanten als Array:** Ein beliebiger Datentyp im Menü `Data` ⇒ `Constant` kann zu einem Array werden, indem Sie `Properties` und anschließend unter `Konfiguration` die Option `1D Array` auswählen. Die Größe kann hier eingegeben werden,

oder Sie können bei der Eingabe von Werten die **Eingabetaste** drücken, um weitere Werte anzuhängen.

- **Benennen von Feldern:** Durch Umbenennen der Eingangsanschlüsse am Objekt `Build Record` können Sie Ihrem Datensatz spezifische Feldnamen zuweisen wie beispielsweise `int`, `rand` und `wave`.
- **Der Eingang "Default Value Control":** Eine `Record Constant` bildet ein hervorragendes interaktives Anzeigeobjekt, wenn Sie einen `Default Value Control`-Pin für die Steuerung der Standardwerte hinzufügen. Die `Record Constant` konfiguriert sich automatisch selbst für den Datensatz, den sie empfängt.

Manipulieren von Datensätzen, Schritt 2

Verwenden Sie einen bedingten Ausdruck in einem `Formula`-Objekt, um den Zufallswert in dem Datensatz zu testen, und zeigen Sie entweder den Wert oder eine Textzeichenfolge an. Wenn der nächste Wert niedriger als 0.5 ist, zeigen Sie den Zufallswert an; geben Sie andernfalls eine Textzeichenfolge “`More than 0.5`" aus. Extrahieren Sie als nächstes nur die Uhrzeit und die Wellenform.

Tipp

Verwenden Sie kein `Formula`-Objekt zum Extrahieren der Uhrzeit und der Wellenform. Zeigen Sie diesen Datensatz mit einem `AlphaNumeric`-Objekt an.

Lösung—Manipulieren von Datensätzen, Schritt 2

Abbildung A-28 zeigt das Manipulieren von Datensätzen, Schritt 2.

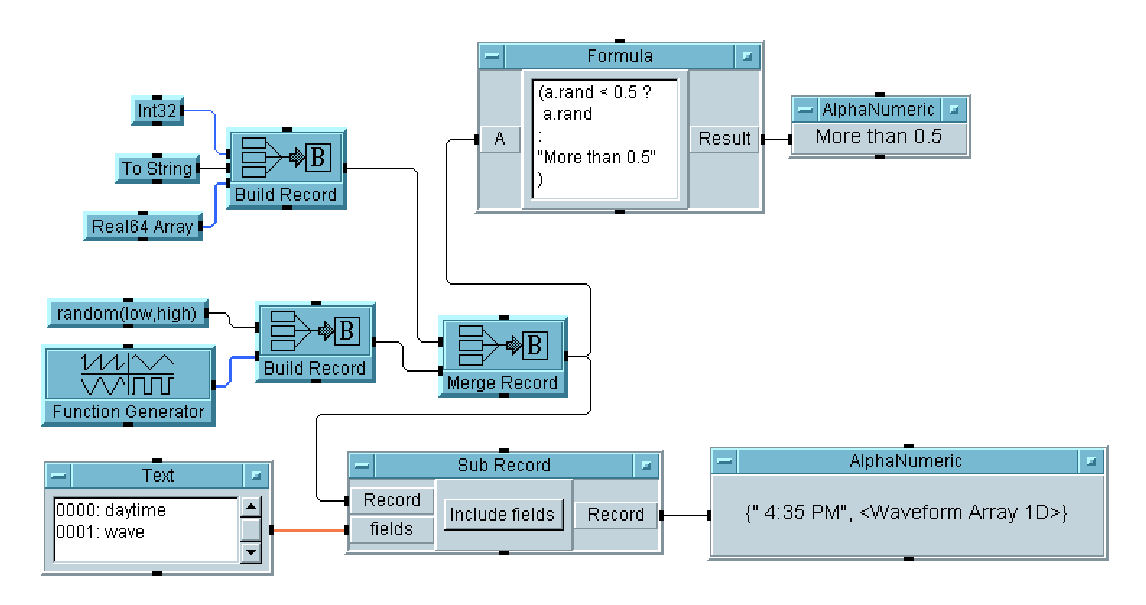


Abbildung A-28. Manipulieren von Datensätzen, Schritt 2

Wichtige Punkte

- **Verwenden eines bedingten Ausdrucks:** VEE unterstützt einen bedingten Ausdruck; dies bietet eine effiziente Möglichkeit, eine If-Then-Else-Aktion zu implementieren. Der bedingte Ausdruck in diesem `Formula`-Objekt wird als "triadischer Ausdruck" bezeichnet. Er lautet `(a.rand < 0.5? a.rand: "More than 0.5")`. Beachten Sie, dass es sich dabei um einen einzigen vollständigen Ausdruck handelt; Sie können diesen Ausdruck mit Zeilenendezeichen im `Formula`-Objekt wir gezeigt schreiben. Wenn das `Formula`-Objekt mehrere Ausdrücke enthält, werden die Ausdrücke durch Semikolons (`;`) getrennt.

- **Das Objekt "Sub Record" (Teildatensatz):** Beachten Sie den `Text`-Array der Felder an den Eingangs-Pins des `Sub Record`. Wenn Sie das Objekt `Sub Record` so konfigurieren, dass es Felder enthält, gibt es einen Datensatz aus, der nur die angegebenen Felder enthält.

Manipulieren von Datensätzen, Schritt 3

Ersetzen Sie den ganzzahligen Eingang für das erste Feld durch ein Objekt `For Count` und durchlaufen Sie 10 Iterationen. Vergewissern Sie sich, dass Sie bei jeder Iteration einen “ping” auf den Zufallszahlen-Generator und auf die Uhrzeitfunktion ausführen. Senden Sie den vollständigen Datensatz an ein Objekt `To DataSet`. Rufen Sie in einem separaten Thread alle Datensätze aus dem Dataset auf, in dem der Zufallswert größer als 0.5 ist. Legen Sie die resultierenden Datensätze in einer Datensatzkonstanten ab.

Tipp

Sie benötigen einen Steuer-Pin für einen `Default Value` (Standardwert) am Objekt `Record Constant` .

Lösung—Manipulieren von Datensätzen, Schritt 3

Abbildung A-29 zeigt eine Lösung zum Manipulieren von Datensätzen, Schritt 3.

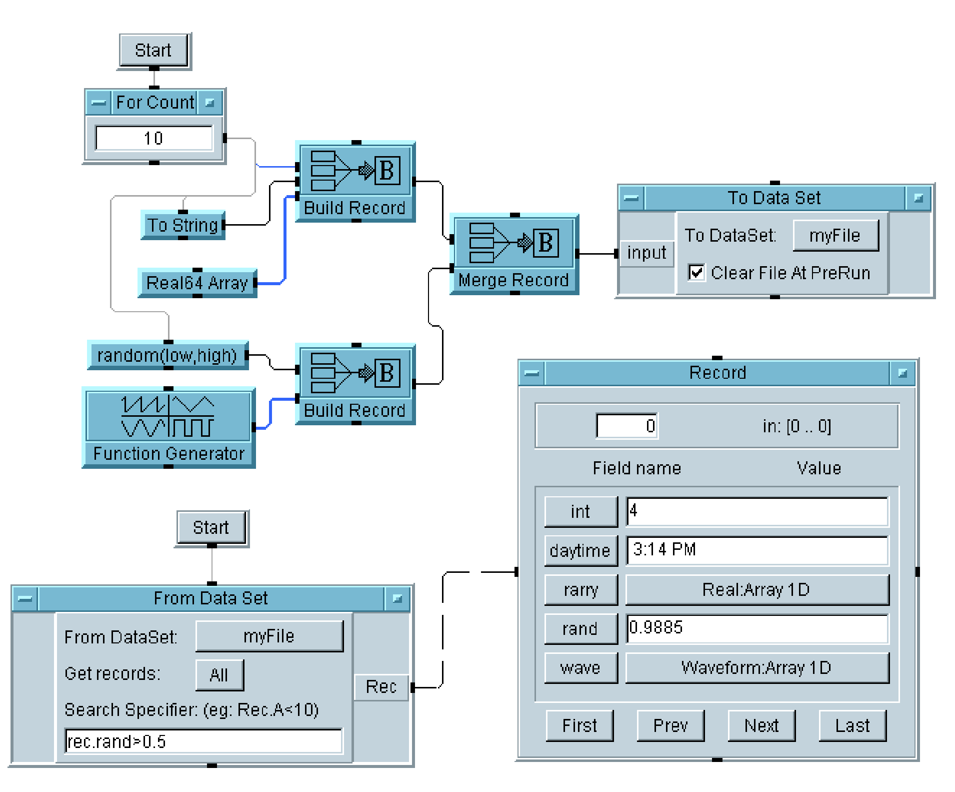


Abbildung A-29. Manipulieren von Datensätzen, Schritt 3

Wichtige Punkte

- **Das Objekt "To DataSet":** Die Option `Clear File at PreRun` löscht nur die Datei, bevor zum ersten Mal Daten gesendet werden. Beachten Sie, dass das Programm 10 verschiedene Datensätze sequenziell an die gleiche Datei sendet, die an die Datei angehängt werden.

- **Das Objekt "From DataSet":** Dieses Objekt wird so konfiguriert, dass es alle Datensätze abruft, bei denen das Feld `rand` größer als 0.5 ist. In diesem Fall erfüllen fünf der zehn Datensätze dieses Kriterium, und der erste Datensatz wird mit der Indexnummer 0 angezeigt,

Sequenzieren von Tests

Verwenden des Test-Sequencer, Schritt 1

Erstellen Sie eine einfache UserFunction mit dem Namen UpperLimit als Einblendfenster mit einem Real64 Slider und einem Objekt Confirm (OK). Senden Sie die Ausgabe des Slider an eine globale Variable mit dem Namen UpLimit und auch an einen Ausgangsanschluss. Erstellen Sie ein Sequencer-Objekt und konfigurieren Sie test1 im Sequencer als EXEC-Transaktion, die UpperLimit aufruft.

Erstellen Sie eine weitere Funktion mit dem Namen AddRand, die den Test simuliert, den Sie aufrufen können. Diese Funktion sollte einen Eingangswert zu einem Zufallswert (0 bis 1) addieren. Sie hat einen Eingangs-Pin und einen Ausgangs-Pin.

Erstellen Sie vom Sequencer aus test2, um AddRand aufzurufen und eine Null zu senden. Testen Sie den Rückgabewert und vergleichen Sie ihn mit einem Grenzwert, der kleiner ist als der globale UpLimit-Wert. Falls der Test erfolgreich ist, geben Sie "PASS" + test2.result zurück. Schlägt der Test fehl, geben Sie "FAILED" + test2.result zurück. Legen Sie eine Alphanumeric-Anzeige am Return-Pin des Sequencer ab.

Führen Sie nach dem Sequencer-Objekt einen "Ping" auf das Objekt Get Variable (UpLimit) und eine weitere Alphanumeric-Anzeige aus. Führen Sie das Programm mehrmals aus.

Lösung—Verwenden des Test-Sequencer, Schritt 1

Abbildung A-30 zeigt eine Lösung für den ersten Schritt zur Verwendung des `Sequencer`.

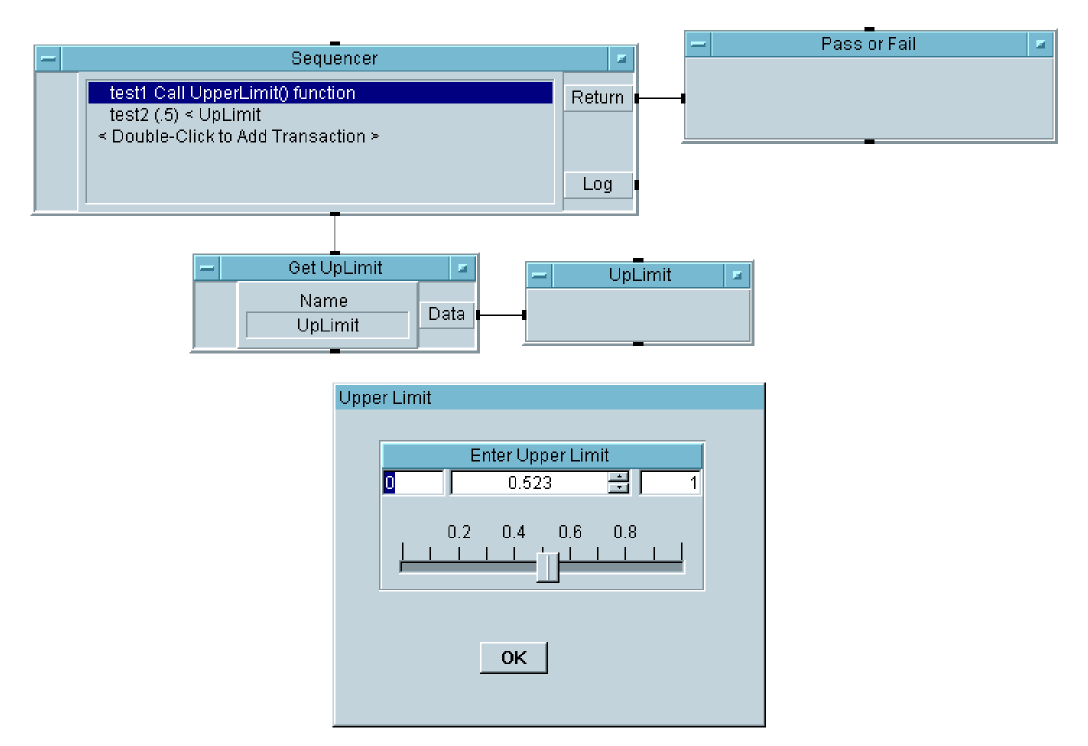


Abbildung A-30. Verwenden des Test-Sequencer, Schritt 1

Wichtige Punkte

- **Einstellen von globalen Variablen mit einer UserFunction:** Eine typische Verwendung der ersten `Sequencer`-Transaktion ist der Aufruf einer `UserFunction`, die die `Globalen Variablen` festlegt wie in diesem Fall. Anschließend können Sie diese Variablen in allen nachfolgenden Tests verwenden, wie hier gezeigt.

- **Der Return-Pin des Sequencer:** Der `Return`-Pin in diesem Beispiel liefert eine Meldung PASS oder FAIL (erfolgreich oder fehlgeschlagen) sowie den Testwert. Sie können mit diesem Pin eine beliebige Meldung und einen Wert aus einem bestimmten Test liefern.

Verwenden des Test-Sequencer, Schritt 2

Inaktivieren Sie den ersten Test. Wenn Sie die globale Variable nicht auch ein einer anderen Stelle benötigen, können Sie die Funktion `UpperLimit` direkt aufrufen. Ändern Sie `test2`, sodass der Rückgabewert von `AddRand(0)` mit dem Ergebnis der Funktion `UpperLimit` verglichen wird.

Tipp: Verwenden Sie zum Inaktivieren des ersten Tests das Fenster `Sequencer Transaction`, wie in Abbildung A-31 gezeigt.

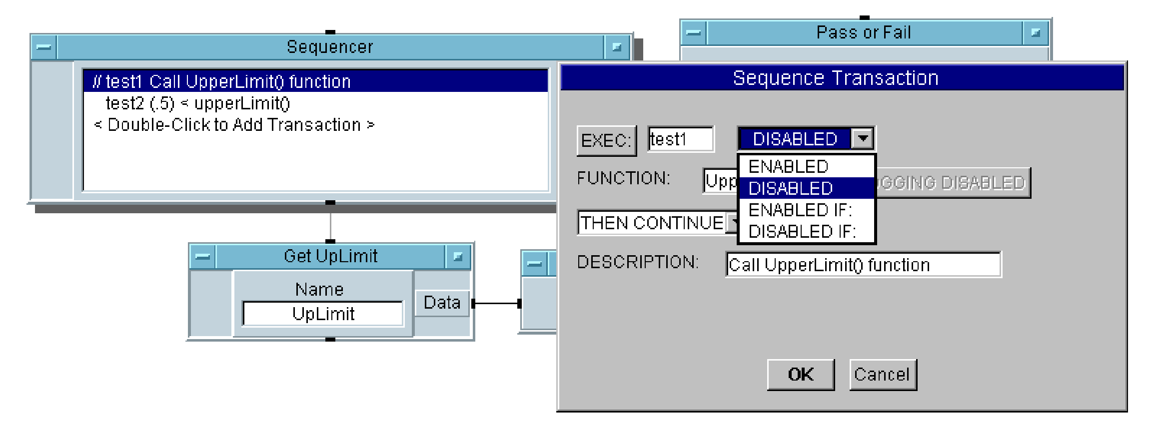


Abbildung A-31. Inaktivieren des ersten Tests in der Sequenz

Beachten Sie, dass in Abbildung A-31 der erste Test im `Sequencer` mit den beiden Schrägstrichen "auskommentiert" und damit inaktiviert ist.

Lösung—Verwenden des Test-Sequencer, Schritt 2

Abbildung A-32 zeigt eine Lösung zur Verwenden des Test-`Sequencer`, Schritt 2.

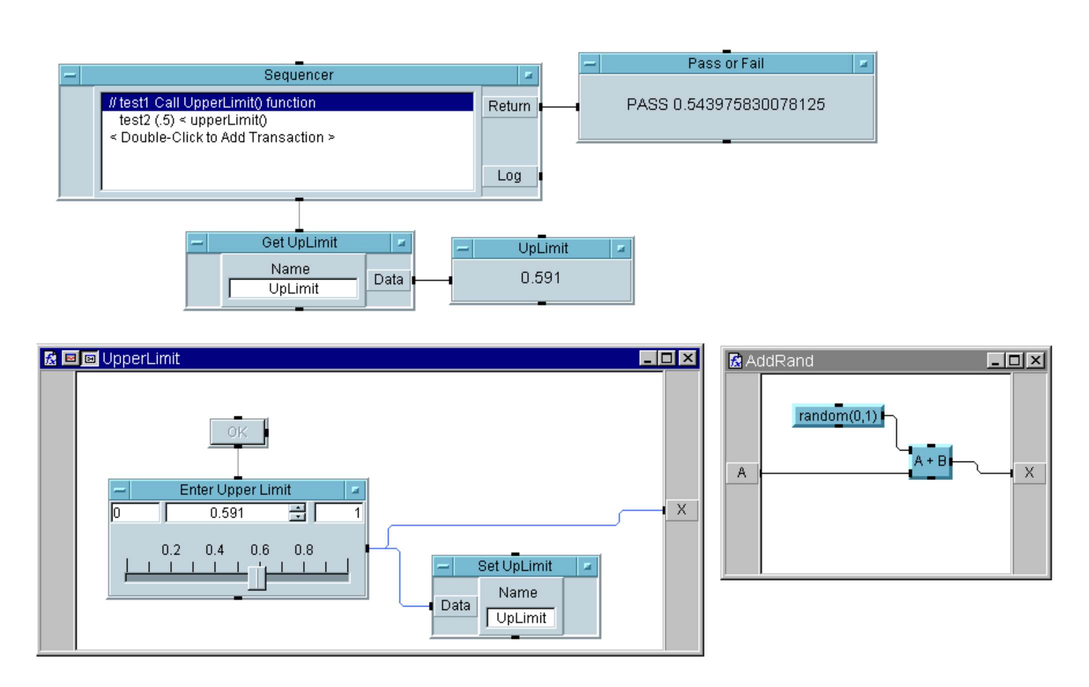


Abbildung A-32. Verwenden des Test Sequencer, Schritt 2

Wichtige Punkte

- **Die UserFunction in einem Ausdrucksfeld:** In diesem Beispiel können Sie, statt ein Testergebnis mit der Variablen `UpLimit` zu vergleichen, den Funktionsnamen `UpperLimit()` in dem Ausdrucksfeld eingeben, in das die Variable geschrieben wird.

Verwenden des Test Sequencer, Schritt 3

Bearbeiten Sie die Transaktion `test2 Sequencer`, die die VEE-Funktion `random (0,1)` aufruft. Vergleichen Sie das Ergebnis mit einem Grenzwert kleiner als `0.5`. Schneiden Sie die Transaktion `test1` aus und fügen Sie sie ein, bis Sie insgesamt vier Tests haben.

Erstellen Sie ein Programm, mit dem der `Sequencer` fünf Mal ausgeführt wird. Zeichnen Sie die Daten in einem Dataset von Datensätzen auf, und erfassen Sie die Daten in einem Array. Ermitteln Sie mit diesem Array den kleinsten und größten Wert, den Mittelwert und die Standardabweichung der Ergebnisse des zweiten Tests.

Lösung—Verwenden des Test Sequencer, Schritt 3

Abbildung A-33 zeigt eine Lösung für Schritt 3.

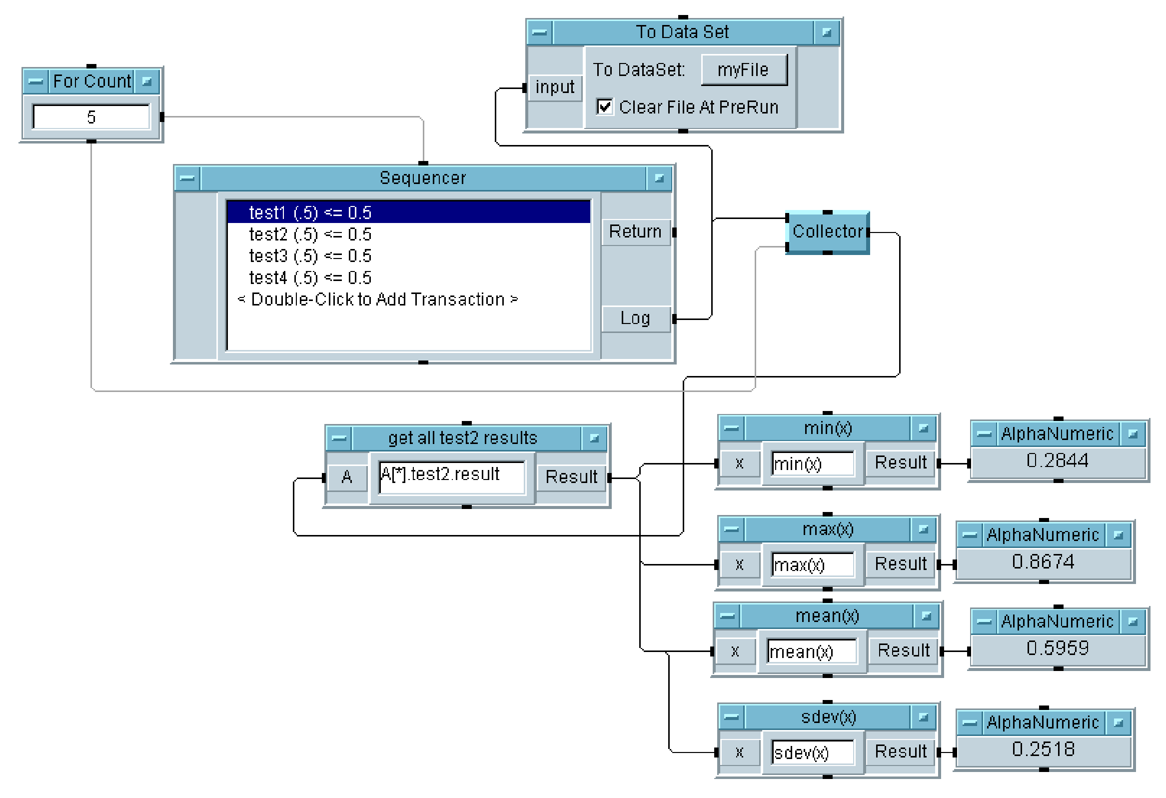


Abbildung A-33. Verwenden des Test Sequencer, Schritt 3

Wichtige Punkte

- **Das Datenformat für mehrere Durchläufe des Sequencer (erster Thread):** Wenn der `Sequencer` ein Mal ausgeführt wird, gibt er einen `Record of Records` (Datensatz aus Datensätzen) aus. Der erste Datensatz enthält Feldnamen, die den Testnamen entsprechen, anschließend enthält jedes Feld einen Datensatz mit den verschiedenen Datenteilen für diesen bestimmten Test. Wenn der `Sequencer` mehrmals ausgeführt wird, kann jeder `Record of Records` einem Array hinzu-

gefügt werden; diese Arrays können anschließend untersucht werden. Wenn Sie das Format *<record>[*].<record>.<field>* im `Formula`-Objekt verwenden, erhalten Sie einen Array aus Daten. In diesem Fall erhalten Sie einen Array aus Real-Werten mit den Testergebnissen aus fünf Durchläufen von `test2`. Sie können anschließend den kleinsten und größten Wert, den Mittelwert und die Standardabweichung aus diesem Array ermitteln. Sie können auch einen einzelnen Durchlauf von `test2` angeben, indem Sie ein bestimmtes Element im Array der Records of Records angeben. Wenn Sie beispielsweise das Ergebnis des ersten Durchlaufs von `test2` erhalten wollen, verwenden Sie den Ausdruck: `A[0].test2.result`.

Verwenden des Test Sequencer, Schritt 4

Fügen Sie dem Aufzeichnungs-Datensatz ein Zeitstempelfeld hinzu. Fügen Sie eine Verzögerung hinzu, sodass jeder Schritt mit einer Sekunde Abstand erfolgt. Rufen Sie in einem separaten Thread alle Ergebnisse von `test2` ab und senden Sie sie an eine Datensatzkonstante.

Tipps

- **Das Objekt "Delay" (Erster Thread):** Dieses Objekt enthält den Ausführungsablauf für die angegebene Anzahl von Sekunden. Hier wird es verwendet, um sicherzustellen, dass sich die Werte der Zeitstempel zwischen den einzelnen Durchläufen des `Sequencer` unterscheiden.

- **Hinzufügen eines Zeitstempels:** Öffnen Sie zum Hinzufügen eines Zeitstempels das `Sequencer`-Objektmenü und wählen Sie das Register `Properties` ⇒ `Logging` aus, um `Record Fields to Log` ⇒ `Time Stamp` zu markieren. Abbildung A-34 zeigt das Register `Properties` ⇒ `Logging`.

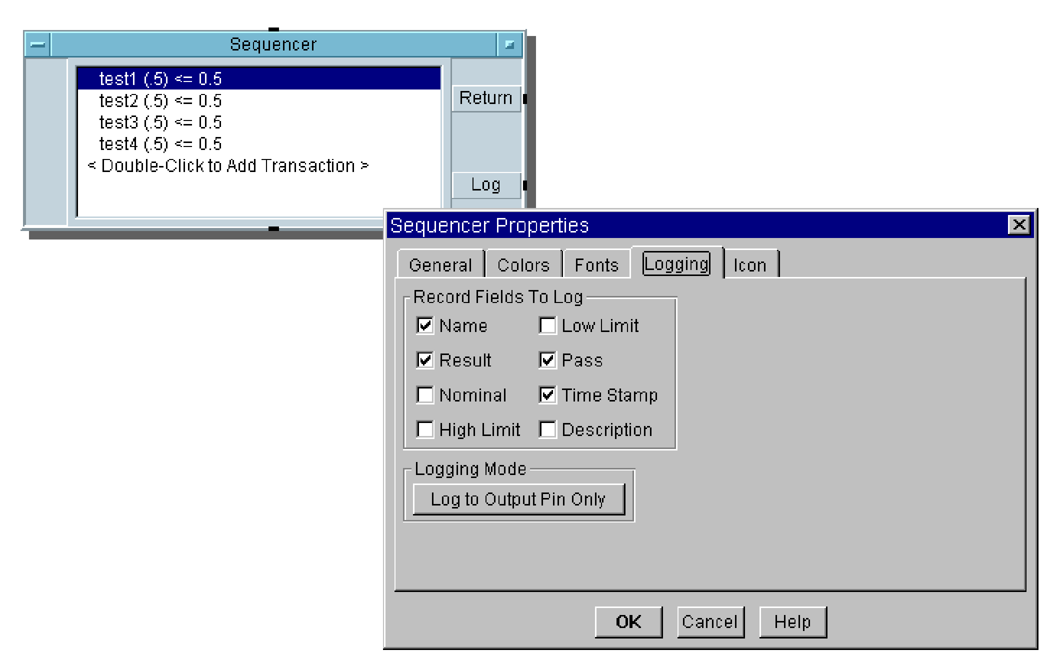


Abbildung A-34. Hinzufügen eines Zeitstempels zum Aufzeichnungs-Datensatz

Lösung—Verwenden des Test Sequencer, Schritt 4

Abbildung A-35 zeigt eine Lösung für Schritt 4 von "Verwenden des Test Sequencer".

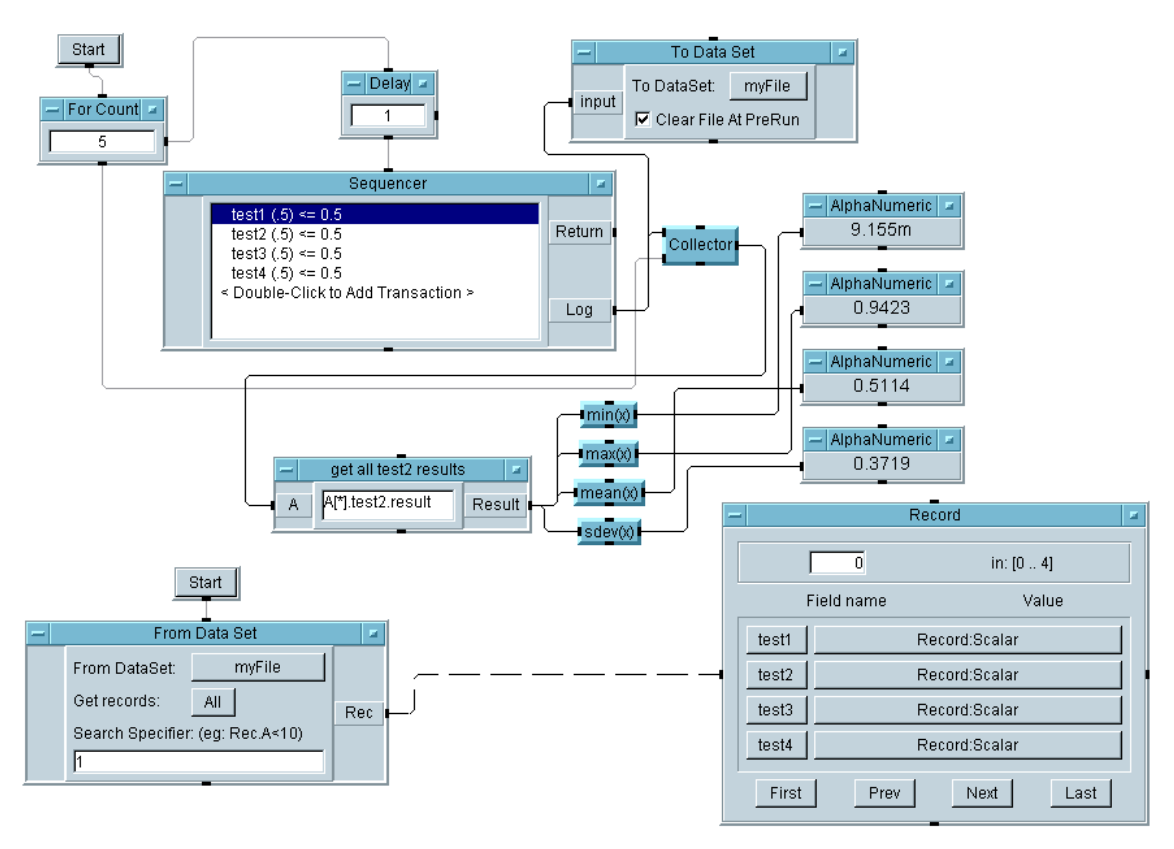


Abbildung A-35. Verwenden des Test Sequencer, Schritt 4

Tipp:

Klicken Sie zum Anzeigen eines Datensatzes das Feld `Record` ⇒ `Record: Scaler` für einen der Tests an, um das Dialogfenster `Record Field Data` anzuzeigen. Abbildung A-36 zeigt das Dialogfenster `Record Field Data`.

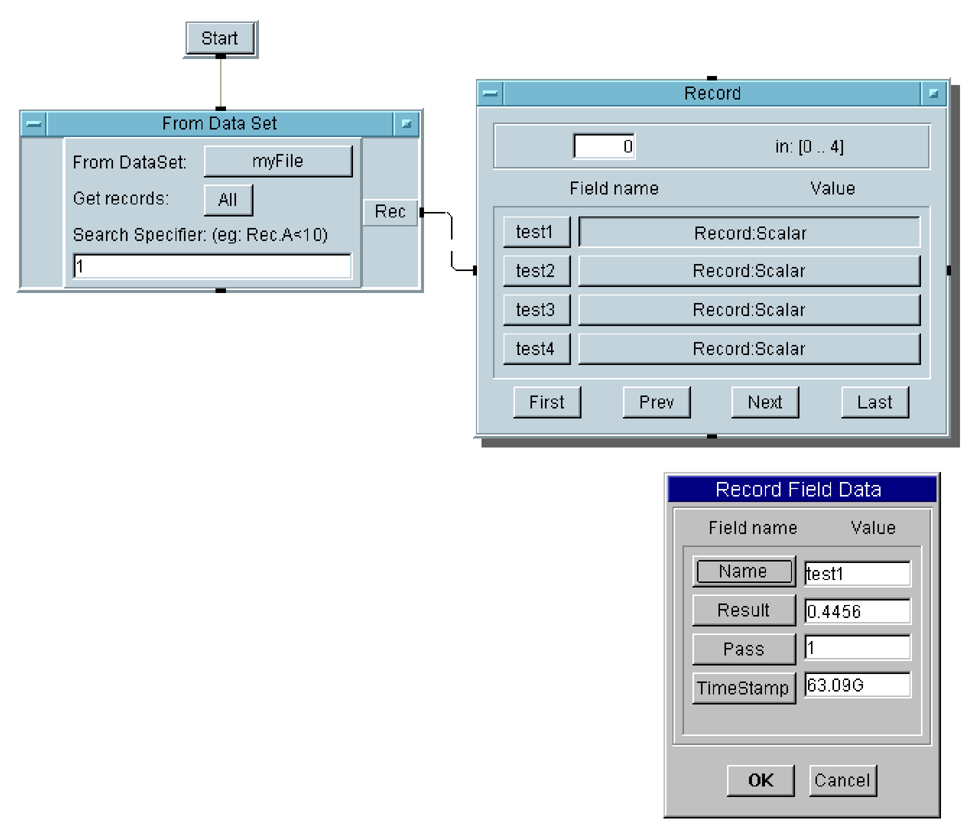


Abbildung A-36. Markieren eines Datensatzes

Verwenden des Test Sequencer, Schritt 5

Geben Sie die Zeitstempelfelder aus den Datensatzfeldern an eine `Logging Alphanumeric`-Anzeige aus.

Tipp: Verwenden Sie vier `Formula`-Objekte (einen für jeden Test). Zum Anzeigen aller vier `Formula`-Ergebnisse in einer `Logging Alphanumeric`-Anzeige fügen Sie ein `Junction`-Objekt hinzu. Verwenden Sie ein Objekt `To String` zum Formatieren des 63G-Zeitstempelwerts in eine leichter lesbare Zeichenfolge.

Lösung—Verwenden des Test Sequencer, Schritt 5

Abbildung A-37 zeigt den Programm-Thread zum Ausgaben der Zeitstempel an eine Anzeige, Schritt 5 von "Verwenden des Test Sequencer".

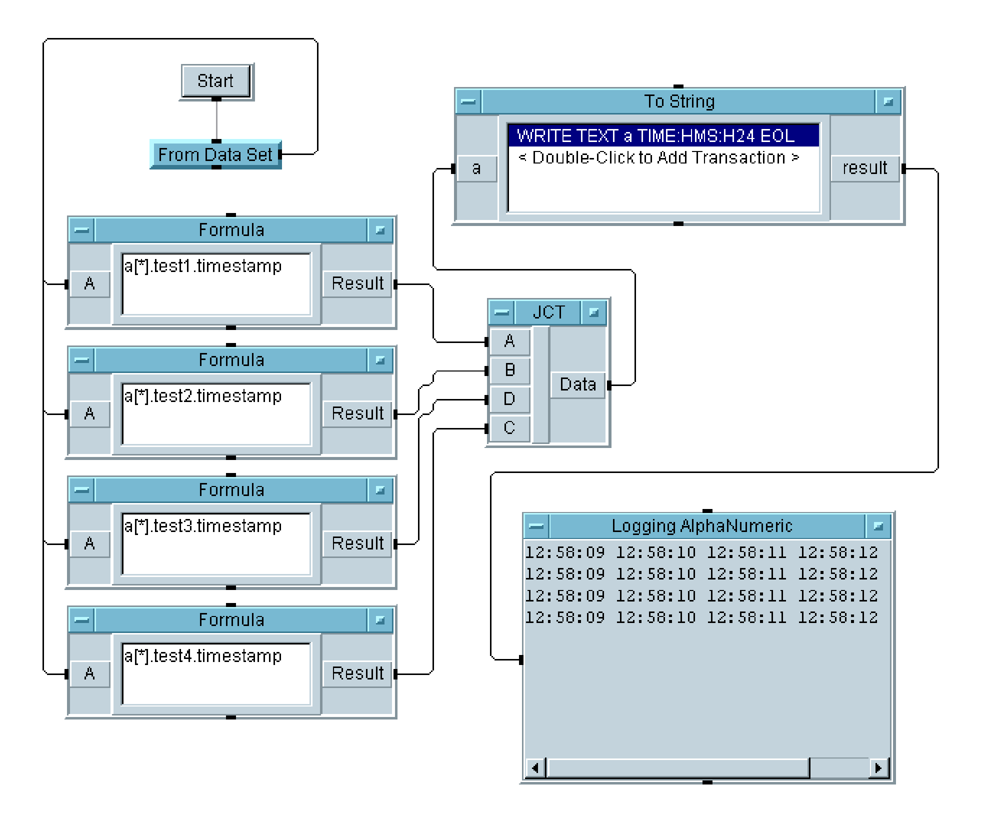


Abbildung A-37. Verwenden des Test Sequencer, Schritt 5

Wichtige Punkte

- **Umwandeln der Zeitstempelformate:** Das Objekt `To String` vor `Logging AlphaNumeric` wandelt die Zeitstempel aus einem `Real`-Format in ein `Time Stamp`-Format um, sodass sie besser lesbar sind.

Verwenden des Test Sequencer, Schritt 6

Wenn der `Sequencer` viele Tests umfasst, kann es sehr aufwändig sein, viele einzelne `Formula`-Objekte mit Verbindung zu einem `Junction`-Objekt zu verwenden. Sie können stattdessen eine `Formula` verwenden, die einen Ausdruck enthält, zur Ausführungszeit einen Ausdruck generieren und eine Schleife durch die möglichen Ausdrücke durchlaufen.

Das Beispiel generiert zunächst die Ausdrucks-Zeichenfolgen.

Verwenden Sie in einem separaten Thread eine Schleife ("Loop") und eine `Formula`, um eine Testausdrucks-Zeichenfolge zu generieren. Geben Sie die Informationen als Zeichenfolge in einer `Logging Alphanumeric`-Anzeige aus. Die in der `Formula` generierte Zeichenfolge sollte `"a[*].test<x>.timestamp"` lauten, wobei `<x>` von 1 bis 4 reicht.

Lösung—Verwenden des Test Sequencer, Schritt 6

Abbildung A-38 zeigt eine Lösung für Schritt 6.

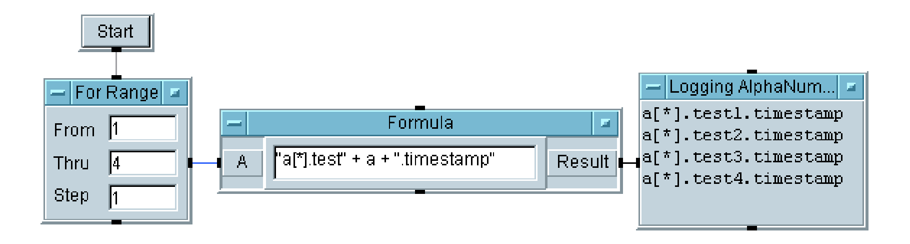


Abbildung A-38. Verwenden des Test Sequencer, Schritt 6

Verwenden des Test Sequencer, Schritt 7

Verwenden Sie jetzt die in Schritt 6 erstellte Schleife und die the `Formula`, und ersetzen Sie die vier `Formulas` und die `Junction` im vorigen Schritt durch die Schleife und `Formula`. Außerdem soll die erstellte Zeichenfolge ausgewertet werden. Senden Sie die generierte Zeichenfolge (den Ausdruck `"a[*].test<x>.timestamp"`) an eine `Formula`, die zur Ausführungszeit ausgewertet werden soll.

Tipps

- **Formula-Steuer-Pin am Formula-Objekt:** Die auszuwertende `Formula` wird von der `Formula` in der Schleife generiert. Sie können ein zweites `Formula`-Fenster mit einem Steuereingang für seinen `Formula`-Ausdruck erstellen. Der Ausdruck, den die zweite `Formula` auswertet, wird zur Ausführungszeit generiert.

Lösung—Verwenden des Test Sequencer, Schritt 7

Abbildung A-39 zeigt eine Lösung für Schritt 7.

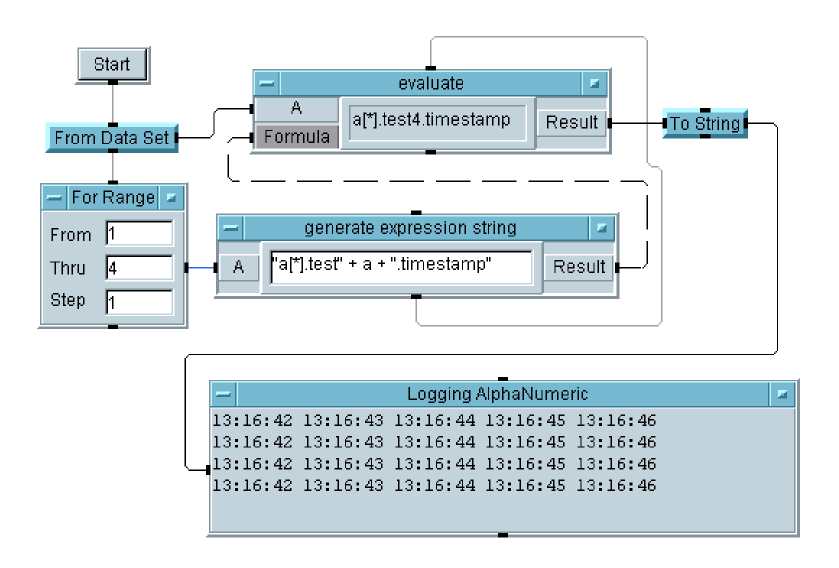


Abbildung A-39. Verwenden des Test Sequencer, Schritt 7

Wichtige Punkte

- Das Objekt `To String` wird noch immer verwendet, um den `Real64`-Wert im Zeitstempelformat zu formatieren.

- Beachten Sie die Sequenzlinie zwischen der ersten `Formula` ("Generierung") und der zweiten `Formula` ("Auswertung"). Auf diese Weise wird sichergestellt, dass die zweite `Formula` nicht ausgeführt wird, bis sie die neue auszuwertende Zeichenfolge erhält.

Verwenden des Test Sequencer, Schritt 8

Zeigen Sie nur die Datensätze an, in denen Test 1 erfolgreich war und Test 2 fehlgeschlagen ist.

Lösung—Verwenden des Test Sequencer, Schritt 8

Abbildung A-40 zeigt eine Lösung für den letzten Schritt beim Verwenden des Test Sequencer.

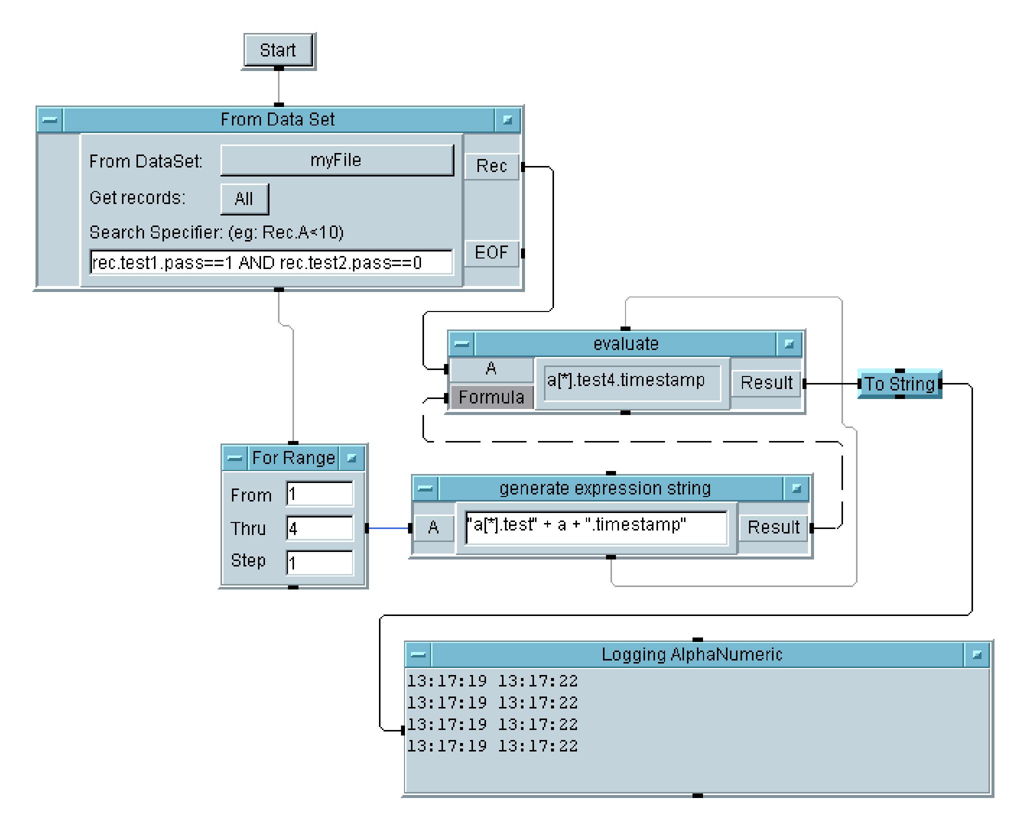


Abbildung A-40. Verwenden des Test Sequencer, Schritt 8

Wichtige Punkte

- **Der EOF-Pin am Objekt "From Data Set" (zweiter Thread):** Der EOF-Pin wird hinzugefügt, falls keine Datensätze vorliegen, die die Kriterien erfüllen. In diesem Fall löst der EOF-Pin aus, statt dass VEE das Programm mit einer Fehlermeldung anhält.

- **Der bedingte Ausdruck im Objekt "From Data Set" (Zweiter Thread):** Der Ausdruck lautet *(Rec.test1.pass==1) OR (Rec.test2.pass== 0)* mit jeweils dem gleichen Format *<Datensatz>.<Datensatz>.<Feld>*. Rec ist der Name der einzelnen Datensätze in dem Dataset beim Lesen und Testen. Test1 und test2 geben an, welche Tests VEE untersuchen soll, und der übergebene Feldname ist der Standardname für den Pass-Fail-Indikator (1 oder 0), der von VEE zugeordnet wird. (Sie aktivieren oder inaktivieren verschiedene Felder für alle Tests durch Auswählen des Registers Logging im Fenster Sequencer Properties.)

Glossar

Glossar

In diesem Glossar werden die im vorliegenden Handbuch definierten Begriffe definiert. Ein vollständiges Glossar der VEE-Begriffe können Sie über `Help` ⇒ `Contents and Index` abrufen. Wählen Sie daraufhin `Reference` und anschließend `Glossary` aus. Klicken Sie in diesem Glossar einen Begriff an, um seine Definition anzuzeigen. Wenn Sie die Definition gelesen haben, klicken Sie eine beliebige Stelle an, um den Text zu löschen.

Abgeblendete Funktion
Eine Menüfunktion, die grau statt schwarz angezeigt wird; dies weist darauf hin, dass die Funktion nicht aktiv oder nicht verfügbar ist. Dialogfensterelemente wie Schaltflächen, Auswahlkästchen oder Radioschaltflächen können ebenfalls abgeblendet sein.

Anklicken
Drücken und Loslassen der Maustaste. Durch Anklicken wird normalerweise eine Menüfunktion oder ein Objekt im VEE-Fenster ausgewählt. Siehe auch **Doppelklicken** und **Ziehen**.

Anschluss
Die interne Darstellung eines Pins, die Informationen zu dem Pin und dem an diesem Pin anliegenden Datencontainer anzeigt. Klicken Sie einen Anschluss doppelt an, um die Containerinformationen anzuzeigen.

Ansichten
VEE zeigt ein Programm in einer von zwei Ansichten an: Die Fensteransicht bietet eine Benutzerschnittstelle zu einem VEE-Programm, und die Detailansicht bietet ein Fenster zur Entwicklung eines VEE-Programms.

Arbeitsbereich
Ein Bereich im `Hauptfenster` (oder den Fenstern `UserObject` und `UserFunction`), in dem Sie VEE-Objekte platzieren und verbinden, um ein VEE-Programm zu erstellen.

Arbeitsbereich

Ein Bereich im VEE-Fenster, das die Programmier- oder Bearbeitungsfenster wie `Hauptfenster`, `UserObject` und `UserFunction` enthält. Diese Fenster enthalten Arbeitsbereiche, in denen Sie VEE-Objekte platzieren und verbinden.

Ausdruck

Eine Gleichung in einem Eingabefeld, die Namen von Eingangsanschlüssen, globalen Variablen, mathematischen Funktionen und benutzerdefinierte Funktionen enthalten kann. Ein Ausdruck wird zur Ausführungszeit ausgewertet. Ausdrücke sind zulässig in den Objekten `Formula`, `If/Then/Else`, `Get Values`, `Get Field`, `Set Field`, `Sequencer` sowie in `Dialogfenster`- und in E/A-Transaktionsobjekten.

Ausschneidepuffer

Der Puffer, der die Objekte enthält, die Sie ausschneiden oder Kopieren. Sie können das Objekt anschließend mit der Schaltfläche `Einfügen` in der Symbolleiste (`Edit` ⇒ `Paste`) wieder in den Arbeitsbereich einfügen.

Auswählen

Auswählen eines Objekts, einer auszuführenden Aktion oder eines Menüelements. Normalerweise wählen Sie etwas mit der Maus aus.

Auswahlfeld

Ein Feld in einem Objekt- oder Dialogfenster, das die Auswahl in einer Dropdown-Liste ermöglicht.

Auswahlkästchen

Ein leicht zurückgesetztes Rechteck in VEE-Menüs und Dialogfenstern, über das Sie eine Einstellung auswählen können. Wenn Sie zum Auswählen das Kästchen anklicken, wird darin ein Häkchen angezeigt, um darauf hinzuweisen, dass eine Auswahl getroffen wurde. Zum Abbrechen der Einstellung klicken Sie das Kästchen einfach erneut an.

Clone

Ein Menüelement in den VEE-Objektmenüs, mit dem Objekte und ihre Verbindungen dupliziert werden. Beim Klonen wird eine Kopie der Objekte und Verbindungen im `Einfüge`puffer abgelegt. Mit `Clone` werden alle Attribute der geklonten Objekte einschließlich Pins, Parameter und Größe kopiert.

Container

Siehe **Datencontainer**.

Cursor

Ein Zeiger (Zirkumflex) in einen Eingabefeld, das anzeigt, wo alphanumerische Daten erscheinen, wenn Sie Informationen über die Tastatur eingeben.

Datencontainer

Das Datenpaket, das über die Leitungen übertragen und von den Objekten verarbeitet wird. Jeder Datencontainer enthält Daten und den Datentyp, die Datenform und die Zuordnungen (sofern vorhanden).

Datenausgangs-Pin

Ein Verbindungspunkt an der rechten Seite eines Objekts, der den Datenfluss an das nächste Objekt weiterleitet und die Ergebnisse der Operation des ersten Objekts an das nächste Objekt weiterleitet.

Dateneingangs-Pin

Ein Verbindungspunkt an der linken Seite eines Objekts, der den Datenfluss in ein Objekt hinein ermöglicht.

Datenfluss

Der Verlauf der Daten durch und zwischen den VEE-Objekten. Daten fließen von links nach rechts durch Objekte; ein Objekt wird jedoch erst ausgeführt, wenn an allen seinen Dateneingangs-Pins Daten anliegen. Daten werden vom Datenausgangs-Pin eines Objekts an den Dateneingangs-Pin des nächsten Objekts weitergegeben. Der Datenfluss ist der wichtigste Faktor, der die Ausführung eines VEE-Programms festlegt.

Datenform

Jeder Datencontainer hat eine Form und einen Typ. Die Datenform kann ein Skalar oder ein Array mit einer oder mehreren Dimensionen sein. In VEE wird ein eindimensionaler Array als "Array 1D" bezeichnet, ein zweidimensionaler Array als "Array 2D" usw.

Datentyp

Jeder Datencontainer hat einen Typ und eine Form. VEE unterstützt eine Vielzahl von Datentypen einschließlich Text, Real64, Real32 und Int32.

Detailansicht

Die Ansicht eines VEE-Programms, die alle Objekte und die Verbindungslinien zeigt.

Direct I/O (Objekt)

Ein Objekt zur Instrumentensteuerung, das VEE eine direkte Steuerung eines Instruments ohne einen Instrumententreiber ermöglicht.

Doppelklicken

Zweimaliges Drücken und Loslassen der Maustaste in schneller Folge. Das Doppelklicken ist meist eine Abkürzung zum Auswählen und Ausführen einer Aktion. Wenn Sie beispielsweise unter `File` ⇒ `Open` doppelt auf einen Dateinamen klicken, wird die Datei ausgewählt und geöffnet.

Dropdown-Liste

Eine Liste von Auswahloptionen, die durch Anklicken des Pfeils auf der rechten Seite eines Auswahlfelds angezeigt werden kann.

Eigenschaften

Objekteigenschaften sind Attribute von VEE-Objekten, die Sie über `Properties` im Objektmenü ändern können, beispielsweise Farben, Schriften und Titel.

Einblendmenü

Ein Menü, das aufgerufen wird, wenn Sie mit der rechten Maustaste klicken. Sie können beispielsweise das Menü `Edit` aufrufen, indem Sie mit der rechten Maustaste einen leeren Bereich im Arbeitsbereich anklicken. Sie können das Objektmenü auch aufrufen, indem Sie mit der rechten Maustaste einen nicht aktiven Bereich des Objekts anklicken.

Eingabefeld

Ein Eingabefeld ist meist ein Teil eines Dialogfensters oder eines änderbaren Objekts und wird zur Eingabe von Daten verwendet. Der Inhalt eines Eingabefelds kann geändert werden, wenn sein Hintergrund weiß ist.

Einstellungen

Einstellungen sind Attribute der VEE-Umgebung, die Sie über die Schaltfläche `Standardeinstelungen` in der Symbolleiste oder über das Menü `File` ⇒ `Default Preferences` ändern können. Sie können beispielsweise die Standardfarben und -schriften sowie das Zahlenformat ändern.

Fensteransicht

Die Ansicht eines VEE-Programms oder eines `UserObject` bzw. einer `UserFunction`, in der nur die Objekte angezeigt werden, die der Benutzer zur Ausführung des Programms und zum Anzeigen der resultierenden Daten benötigt. Sie können mithilfe von Fensteransichten eine Bedienerschnittstelle für Ihr Programm erstellen.

Font

VEE ermöglicht das Ändern der Schriftgröße sowie des Typs zur Anzeige für verschiedene VEE-Objekte, Titel etc.

Gruppenfenster

Ein Gruppenfenster in Microsoft Windows ist ein Fenster, das Symbole für eine Gruppe von Anwendungen enthält. Jedes Symbol startet eine Anwendung in der Gruppe.

Hauptfenster

Ein Fenster, das den primären Arbeitsbereich enthält, in dem Sie ein VEE-Programm entwickeln. Der Arbeitsbereich für dieses Fenster befindet sich im Arbeitsbereich für das VEE-Fenster.

HP-UX

Das von der Hewlett-Packard Company entwickelte Derivat des UNIX-Betriebssystems.

Hypertext

Ein System zum Verbinden von Themen, über das Sie zu einem verwandten Thema springen können, wenn Sie weitere Informationen aufrufen wollen. In Online-Hilfesystemen werden Hypertext-Verbindungen normalerweise durch unterstrichenen Text gekennzeichnet. Wenn Sie einen solchen Text anklicken, werden die zugehörigen Informationen angezeigt.

In Symbolgröße, Schaltfläche

Eine Schaltfläche in einem Objekt oder im VEE-Fenster, über die das Objekt bzw. das VEE-Fenster auf Symbolgröße verkleinert wird.

In Symbolgröße, Schaltfläche

Eine Schaltfläche in einem `UserObject`, einer `UserFunction` oder im `Hauptfenster`, mit der das `UserObject`, die `UserFunction` oder das `Hauptfenster` auf den gesamten verfügbaren Arbeitsbereich ausgedehnt werden kann.

Kaskadierendes Menü

Ein Untermenü eines Pulldown- oder Einblendmenüs, das weitere Auswahlmöglichkeiten bietet.

Komponente

Eine einzelne Instrumentenfunktion oder ein Messwert in einem VEE-Instrumentenfenster oder einem Komponententreiber. Ein Treiber für einen Voltmeter enthält beispielsweise Komponenten, die den Bereich, die Auslöserquelle und den letzten Lesevorgang aufzeichnen.

Komponententreiber

Ein Instrumenten-Steuerobjekt, das Werte aus speziell ausgewählten Komponenten schreibt oder in diese Komponenten schreibt. Verwenden Sie Komponententreiber zum Steuern eines Instruments mit einem Treiber durch Setzen der Werte weniger Komponenten zu einem Zeitpunkt. (Komponententreiber unterstützen keine Kopplung.)

Kontext

Eine Ebene des Arbeitsbereichs, die weitere Ebenen der Arbeitsbereiche wie beispielsweise verschachtelte `UserObjects` enthalten kann, von diesen jedoch unabhängig ist.

Menüleiste

Die Leiste am oberen Rand des VEE-Fensters, das die Titel der Pull-down-Menüs anzeigt, aus denen Sie Befehle und Objekte auswählen können.

Objekt

i. Ein grafische Darstellung eines Elements in einem Programm, z. B. eines Instruments, eines Steuerelements, einer Anzeige oder eines mathematischen Operators. Ein Objekt wird im Arbeitsbereich platziert und mit anderen Objekten verbunden, um ein Programm zu erstellen.

ii. Ein Datentyp für die ActiveX-Automatisierung und -Steuerelemente.

Objektmenü

Das einem Objekt zugeordnete Menü, das Funktionen enthält, die mit dem Objekt ausgeführt werden (z. B. Verschieben, Größe ändern, Kopieren und Löschen des Objekts). Klicken Sie zum Abrufen des Objektmenüs die Schaltfläche "Objektmenü" in der linken oberen Ecke des Objekts an, oder klicken Sie mit der rechten Maustaste, während sich der Zeiger auf dem Objekt befindet.

Offene Ansicht

Die Darstellung eines VEE-Objekts, die mehr Details als die Symbolansicht anzeigt. Die meisten Objekte zeigen in der offenen Ansicht Felder an, über die die Arbeitsweise des Objekts geändert werden kann.

Panel-Treiber

Ein Objekt zur Steuerung, das eine Abstimmung aller Funktionseinstellungen im entsprechenden physischen Instrument mit den Einstellungen des Steuerfensters in der offenen Ansicht des Objekts erzwingt.

Pin (oder Pins)

Ein externer Verbindungspunkt an einem Objekt, an dem eine Linie angeschlossen werden kann.

Programm

In VEE ein grafisches Programm, das aus einer Reihe von Objekten besteht, die mit Linien verbunden sind. Das Programm stellt normalerweise eine Lösung zu einem Engineering-Problem dar.

Program Explorer

Eine Einrichtung im VEE-Fenster, die das Untersuchen eines Programms ermöglicht, insbesondere der Teile eines Programms, die auf dem physischen Bildschirm nicht sichtbar sind.

Pulldown-Menü

Ein Menü, das aus der Menüleiste aufgeklappt wird, wenn Sie mit dem Zeiger auf einen Menütitel zeigen und mit der linken Maustaste klicken.

Rollpfeil

Ein Pfeil, der nach dem Anklicken durch eine Liste von Datendateien oder anderen Auswahloptionen im Dialogfenster blättert oder den Arbeitsbereich verschiebt.

Schaltfläche

Ein grafisches Objekt in VEE, das einen beweglichen Schalter oder eine Auswahltaste simuliert. Schaltflächen werden in der Anzeige als leicht erhöhte Tasten dargestellt. Wenn Sie eine Schaltfläche in VEE "drücken" bzw. mit der Maus anklicken, wird eine Aktion ausgeführt.

Schaltfläche "Objektmenü"

Die Schaltfläche in der linken oberen Ecke eines Objekts in der offenen Ansicht; beim Anklicken dieser Schaltfläche wird das Objektmenü geöffnet.

Schiebeleiste

Ein rechteckiger Balken, der beim Ziehen durch eine Liste von Datendateien oder anderen Auswahloptionen im Dialogfenster blättert oder den Arbeitsbereich verschiebt.

Sequenzausgangs-Pin

Der *untere* Pin an einem Objekt. Wenn dieser Pin angeschlossen ist, wird er aktiviert, sobald die Ausführung des Objekts und die gesamte Datenweitergabe von diesem Objekt abgeschlossen sind.

Sequenzeingangs-Pin

Der *obere* Pin an einem Objekt. Wenn dieser Pin angeschlossen ist, wird die Ausführung des Objekts so lange angehalten, bis der Pin einen Container empfängt (bis ein "Ping" ausgeführt wird).

Statusfeld

Ein Feld mit Informationen, die nicht geändert werden können. Ein Statusfeld sieht aus wie ein Eingabefeld, hat jedoch einen grauen Hintergrund.

Statusleiste

Eine Zeile am unteren Rand des VEE-Fensters, in der Informationen zum aktuellen Status und zu VEE angezeigt werden.

Symbol

1. Ein kleine, grafische Darstellung eines VEE-Objekts, beispielsweise eines Instruments, eines Steuerelementes oder einer Anzeige.

2. Ein kleine, grafische Darstellung einer Anwendung, einer Datei oder eines Ordners im Betriebssystem Microsoft Windows und HP-UX (mit VUE).

Symbolleiste

Die rechteckige Leiste am oberen Rand des VEE-Fensters, die Schaltflächen für den schnellen Zugriff auf häufig verwendete Befehle enthält. Über die Schaltflächen werden Befehle in Menüs wie beispielsweise `File`, `Edit`, `View`, `Device` und `Debug` ausgeführt.

Titelleiste

Die rechteckige Leiste am oberen Rand der offenen Ansicht eines Objekts oder Fensters, die den Titel des Objekts oder Fensters anzeigt. Sie können die Titelleiste eines Objekts über `Properties` im Objektmenü ausschalten.

Transaktion

Die Spezifikationen für die von bestimmten Objekten in VEE verwendete Ein- und Ausgabe (I/O). Beispiele hierzu sind die Objekte `To File`, `From File`, `Sequencer` und `Direct I/O`. Transaktionen erscheinen in der offenen Ansicht dieser Objekte als Texte.

UserObject

Ein Objekt, das eine Gruppe von Objekten zur Ausführung eines bestimmten Zwecks in einem Programm einkapseln kann. Ein UserObject ermöglicht die Anwendung von Top-Down-Design-Techniken beim Erstellen eines Programms und das Erstellen von benutzerdefinierten Objekten, die in einer Bibliothek gespeichert und wiederverwendet werden können.

Weitergabe

Die Regeln, nach denen Objekte und Programme arbeiten bzw. ausgeführt werden. Siehe auch **Datenfluss**.

Windows 95, Windows 98, Windows NT 4.0, Windows 2000

Von Microsoft Corporation entwickelte Betriebssysteme, unter denen VEE ausgeführt werden kann.

Zeiger

Das grafische Abbild, das der Bewegung der Maus zugeordnet ist. Der Zeiger ermöglicht das Auswählen und bietet Rückmeldungen zu einem bestimmten aktiven Prozess. VEE enthält Zeiger mit verschiedenen Formen für den jeweiligen Verarbeitungsmodus, z. B. Pfeil, Fadenkreuz oder Sanduhr.

Ziehen

Drücken und *gedrückt Halten* einer Maustaste, während die Maus bewegt wird. Durch Ziehen können Komponenten (z. B. ein Objekt oder eine Verschiebeleiste) verschoben werden.

Index

B

C

D

G

H

N

O

P

R

S

T

U

V

W

X

Z